PENERAPAN NER-LDA BERBASIS MODEL *TRANSFORMER* UNTUK DETEKSI DAN KLASIFIKASI *TWEET* KEBIJAKAN PUBLIK DI INDONESIA

(Skripsi)

Oleh

DINA MERLIA EFENDI

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025

ABSTRACT

IMPLEMENTATION NER-LDA BASED TRANSFORMER MODEL FOR DETECTION AND CLASSIFICATION OF PUBLIC POLICY TWEETS IN INDONESIA

By

Dina Merlia Efendi

The integration of technological innovation has significantly transformed communication structures and shifted daily life toward digitalization. One of the most popular social media platforms in Indonesia is X (Twitter). Tweets can serve as valuable information for the government and stakeholders to optimize public policy. Therefore, an effective method is needed to detect and classify public concerns expressed in trending policy-related tweets on X. This study applies Named Entity Recognition – Latent Dirichlet Allocation (NER-LDA) to optimize text classification using Transformer-based models, including Indonesian Bidirectional Encoder Representations from Transformers (IndoBERT), Distilled Bidirectional Encoder Representations from Transformers Multilingual (DistilBERT Multilingual), and Cross-lingual Language Model-Robustly Optimized Bidirectional Encoder Representations From Transformers Pretraining Approach (XLM-RoBERTa). The study also implements data augmentation using Large Language Model Meta AI (LLaMA) and hyperparameter tuning using Grid Search. The best parameter combination for the three Transformer models during trending data was a batch size of 16 and a learning rate of 1×10^{-5}. For post trending data, the optimal batch size was 32 for IndoBERT and 16 for DistilBERT Multilingual and XLM-RoBERTa, with the same optimal learning rate of 1×10^{-5} across all models. The comparison of the three Transformer models shows that the best evaluation result for during trending data was achieved by the DistilBERT Multilingual model, while for post trending data was achieved by IndoBERT, with both models reaching an accuracy of 99%. These findings demonstrate that Transformer-based models perform optimally in classifying public policy tweets, both during and post trending periods.

Keywords: DistilBERT Multilingual, IndoBERT, NER-LDA, Transformer, XLM-RoBERTa.

ABSTRAK

PENERAPAN NER-LDA BERBASIS MODEL *TRANSFORMER* UNTUK DETEKSI DAN KLASIFIKASI *TWEET* KEBIJAKAN PUBLIK DI INDONESIA

Oleh

Dina Merlia Efendi

Integrasi inovasi teknologi signifikan mengubah tatanan komunikasi mentransformasikan kehidupan ke arah digital. Platform media sosial populer di Indonesia salah satunya X (Twitter). Ruang lingkup *tweet* dapat menjadi informasi sebagai acuan pemerintah dan pemangku kepentingan mengoptimalkan kebijakan di Indonesia. Oleh karena itu, diperlukan cara yang tepat untuk mendeteksi dan mengklasifikasi *tweet* keresahan masyarakat terhadap fenomena kebijakan publik yang *trending* pada platform X. Penelitian ini menerapan *Named Entity Recognition – Latent Dirichlet Allocation* (NER-LDA) dalam mengoptimalkan klasifikasi teks menggunkan model *Transformer* yang dibandingkan mulai dari *Indonesian Bidirectional Encoder Representations from Transformers* (IndoBERT), *Distilled Bidirectional Encoder Representations from Transformers Multilingual* (DistilBERT *Multilingual*), hingga *Cross-lingual Language Model-Robustly Optimized Bidirectional Encoder Representations From Transformers Pretraining Approach* (XLM-RoBERTa). Penelitian ini menerapkan augmentasi data *Large Language Model Meta AI* (LLaMA) dan *hyperparameter tuning Grid Search.* Hasil kombinasi parameter terbaik pada ketiga model *Transformer* dari data saat *trending* adalah *batch size* 16 dan *learning rate* sebesar 1×10^{-5} lalu pada data setelah *trending* adalah *batch size* 32 untuk model IndoBERT serta *batch size* 16 untuk model DistilBERT *Multilingual* dan XLM-RoBERTa dengan *learning rate* terbaik untuk ketiga model sebesar 1×10^{-5}. Perbandingan ketiga model *Transformer* memperoleh hasil evaluasi terbaik untuk data saat *trending* adalah model DistilBERT *Multilingual* dan data setelah *trending* adalah model IndoBERT dengan *accuracy* kedua model mencapai 99%. Hasil menunjukkan bahwa model *Transformer* memiliki kinerja optimal dalam klasifikasi *tweet* kebijakan publik, baik pada data saat maupun setelah *trending*.

Kata kunci: DistilBERT *Multilingual*, IndoBERT, NER -LDA, Transformer, XLM-RoBERTa.

PENERAPAN NER-LDA BERBASIS MODEL *TRANSFORMER* UNTUK DETEKSI DAN KLASIFIKASI *TWEET* KEBIJAKAN PUBLIK DI INDONESIA

Oleh

Dina Merlia Efendi
2117031089

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025

Judul Skripsi	: **PENERAPAN NER-LDA BERBASIS MODEL *TRANSFORMER* UNTUK DETEKSI DAN KLASIFIKASI *TWEET* KEBIJAKAN PUBLIK DI INDONESIA**
Nama Mahasiswa	: Dina Merlia Efendi
Nomor Pokok Mahasiswa	: **2117031089**
Program Studi	: **Matematika**
Fakultas	: **Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dr. Dian Kurniasari, S.Si., M.Sc.	**Favorisen R. L, S.Kom., M.Si., Ph.D.**
NIP. 196903051996032001	NIP. 198301102008121002

2. Ketua Jurusan Matematika

Dr. Aang Nuryaman, S.Si., M.Si.
NIP. 197403162005011001

MENGESAHKAN

1. Tim Penguji

   Ketua : **Dr. Dian Kurniasari, S.Si., M.Sc.** .....

   Sekretaris : **Favorisen R. L, S.Kom., M.Si., Ph.D.** .....

   Penguji
   Bukan Pembimbing : **Ir. Warsono, M.S., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 19711001200501102

Tanggal Lulus Ujian Skripsi: **10 Juni 2025**

PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama	: **Dina Merlia Efendi**
Nomor Pokok Mahasiswa	: **2117031089**
Jurusan	: **Matematika**
Judul Skripsi	: **PENERAPAN NER-LDA BERBASIS MODEL *TRANSFORMER* UNTUK DETEKSI DAN KLASIFIKASI *TWEET* KEBIJAKAN PUBLIK DI INDONESIA**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah karya penulisan ilmiah Universitas Lampung.

Bandar Lampung, 17 Juni 2025
Penulis

Dina Merlia Efendi
NPM. 2117031089

RIWAYAT HIDUP

Dina Merlia Efendi lahir di Kota Bandar Lampung pada 1 Maret 2004. Anak perempuan kedua dari 3 bersaudara dari pasangan Bapak Solehan Efendy dan Ibu Mariana.

Memulai pendidikan pertama di Taman Kanak-Kanak (TK) Pratama 1 pada tahun 2007-2009 dan memasuki Sekolah Dasar (SD) Negeri 1 Sawah Lama pada tahun 2009-2015. Kemudian menempuh pendidikan di Sekolah Menengah Pertama (SMP) Negeri 1 Bandar Lampung pada tahun 2015-2018 dan Sekolah Menengah Atas (SMA) Negeri 1 Bandar Lampung pada tahun 2018-2021. Selanjutnya, meneguhkan pilihan untuk menempuh pendidikan S1 dengan jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung (Unila) pada tahun 2021.

Dalam dunia akademik selama menjadi mahasiswa saya aktif tergabung dalam organisasi kemahasiswaan di antaranya sebagai berikut:

1. Anggota bidang Eksternal Himpunan Mahasiwa Matematika (HIMATIKA) FMIPA Unila.pada tahun 2023.
2. Anggota divisi Media dan Informasi Badan Eksekutif Mahasiswa (BEM) FMIPA Unila pada tahun 2023.

Selain aktif dalam organisasi, penulis juga aktif mengikuti dan memenangkan berbagai perlombaan di antaranya sebagai berikut:

1. Juara Harapan 1, Kompetisi Infografis Statistik Nasional yang diselenggarakan Universitas Gadjah Mada (UGM) pada Februari 2024.

2. Finalis, Karya Tulis Ilmiah Al-Quran Musabaqah Tilawatil Quran (MTQMN) yang diselenggarakan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (KEMDIKBUDRISTEK) pada November 2024.
3. Juara 2, Kompetisi Karya Tulis Ilmiah Nasional yang diselenggarakan Unila pada September 2023.
4. Finalis, Karya Tulis Ilmiah Nasional Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi (GEMASTIK) yang diselenggarakan KEMDIKBUDRISTEK pada September 2024.
5. Juara 3, Kompetisi Karya Tulis Ilmiah Nasional yang diselenggarakan Universitas Yogyakarta (UNY) pada April 2023.
6. Juara 3, Esai Nasional yang diselenggarakan *English National Competition* (ENC) Universitas Muhammadiyah Sumatera Utara (UMSU) pada Maret 2023.
7. Juara 1, Kompetisi Esai Nasional yang diselenggarakan Universitas Pembangunan Nasional Veteran Jakarta (UPNVJ) pada Agustus 2022.
8. Juara 1, Kompetisi Esai Nasional yang diselenggarakan DEMA FTIK Universitas Islam Negeri (UIN) Prof. K.H. Saifuddin Zuhri Purwokerto pada September 2022.

Salah satu karya tulis hasil gagasan penulis, tim, dan dosen pembimbing berhasil didaftarkan Hak Kekayaan Intelektual (HAKI) dengan judul Model Hybrid VAR-LSTM: Peningkatan Akurasi Peramalan Harga Saham Konstruksi-Kurs Rupiah. Berbagai perolehan prestasi yang penulis dapatkan mengantarkan penulis memperoleh predikat Mahasiswa Berprestasi Ke-1 FMIPA Unila tahun 2024.

Selain itu, penulis mengikuti studi independen dan pengalaman professional menerapkan ilmu yang diperoleh selama perkuliahan di antaranya sebagai berikut:

1. Magang Bersertifikat di Kantor Pusat PT XL Axiata Tbk Kota Jakarta Selatan sebagai *Trade and Network Analytics* pada 2024.
2. Magang di Kantor Pusat Kalbe Consumer Health sebagai *Data Analyst* Kota Jakarta Utara pada 2024.
3. Kerja Praktik (KP) di Badan Riset dan Inovasi Nasional (BRIN) Pusat Riset Lingkungan dan Teknologi Bersih Kota Bandung pada 2024.

KATA INSPIRASI

"Sesungguhnya beserta kesulitan ada kemudahan."
(QS. Al-Insyirah : 7)

"All big things come from small beginnings."
(James Clear)

"All models are wrong, but some are useful."
(George E. P. Box)

"If we always give our best in everything we do, we will be able to break through the limitations holding us back, accomplish our goal, and bring out our full potential."
(Dina Merlia Efendi)

PERSEMBAHAN

Dengan mengucap syukur kehadirat Allah SWT yang telah memberikan taufik dan hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan sukacita yang tulus, saya persembahkan karya ini kepada:

Ayah dan Ibu Tercinta

Yang senantiasa bekerja dengan keras agar penulis dapat menempuh pendidikan dan meraih gelar sarjana, yang tidak kenal lelah untuk selalu mendoakan, memberikan dukungan, nasehat dan kasih sayang yang tidak mungkin terbalas oleh apapun.

Kakak dan Adik Tersayang

Yang telah memberikan semangat, motivasi, doa dan dukungan.

Dosen Pembimbing dan Penguji

Yang senantiasa meluangkan waktu untuk mengarahkan dan memotivasi penulis.

Almamater Tercinta, Universitas Lampung

SANWACANA

Dengan mengucapkan puji dan syukur ke hadirat Tuhan Yang Maha Esa, akhirnya penulis berhasil menyelesaikan skripsi berjudul “Penerapan NER-LDA berbasis model *Transformer* untuk deteksi dan klasifikasi *tweet* kebijakan publik di Indonesia". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Matematika (S.Mat.) di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Dengan penuh rasa syukur, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku Dosen Pembimbing I, terima kasih atas bimbingan, arahan, dukungan, dan kesediaan waktunya selama penyusunan skripsi ini.
2. Bapak Favorisen Rosyking Lumbanraja, S.Kom., M.Si., Ph.D. selaku Dosen Pembibing II, terima kasih untuk segala bantuan dan masukannya selama penyusunan skripsi ini.
3. Bapak Ir. Warsono, M.S., Ph.D. selaku Dosen Penguji, terima kasih atas kesediannya untuk menguji, memberikan saran dan kritik yang membangun dalam penyelesaian skripsi ini.
4. Ibu Dr. Fitriani, S.Si., M.Sc. selaku Pembimbing Akademik, terima kasih untuk setiap bimbingan dan pembelajarannya selama masa perkuliahan.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Seluruh Dosen dan Karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

8. Ayah dan Ibu tersayang yang selalu mengerti, menyemangati, dan mengingatkan penulis untuk tetap semangat dalam menyelesaikan tugas akhir serta doa yang selalu terlimpah untuk keberhasilan penulis.
9. Kakak dan Adik yang telah ikhlas memberikan doa dan perhatian kepada penulis.
10. Almamater tercinta Universitas Lampung.
11. Seluruh pihak yang telah membantu yang tidak dapat disebutkan satu persatu.

Bandar Lampung, 17 Juni 2025
Penulis

Dina Merlia Efendi
NPM. 2117031089

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Integrasi inovasi teknologi signifikan mengubah tatanan komunikasi mentransformasikan kehidupan ke arah digital. Integrasi tersebut mendorong penggunaan media sosial semakin masif. Menurut Badan Pusat Statistik (2024), jumlah penduduk Indonesia pada tahun 2024 adalah 281,6 juta jiwa meningkat hingga 2,9 juta jiwa dibanding periode yang sama pada tahun sebelumnya. Data pengguna media sosial pada awal tahun 2024 di Indonesia mencapai 139 juta pengguna, dibanding dengan total populasi angka tersebut setara 49,9% (We Are Social, 2024). Platform media sosial populer di Indonesia adalah Whatsapp, Instagram, Tiktok, Facebook, dan X (Twitter).

Platform X merupakan media sosial yang mengakomodasi individu berbagi informasi dan aspirasi *real-time*. Pada tahun 2023 Indonesia menempati jumlah pengguna X terbanyak ke-5 di dunia, posisinya di bawah Amerika Serikat, Jepang, India, dan Brazil (Statista Research Department, 2024). Pengguna X di Indonesia mencapai 24,69 juta setara dengan 8,9% total populasi pada awal tahun 2024. Hal tersebut menunjukkan bahwa X menyediakan akses penting berisi *big data* berupa target analisis dinamika kebijakan publik (Asmara dan Butsi, 2020).

Implementasi tanda pagar atau *hashtag* (#) di platform X memfasilitasi pengguna memfokuskan pembicaraan pada suatu topik (Geni dkk., 2023). Pada permasalahan kebijakan publik, beberapa tanda pagar seperti #PeringatanDarurat, #DaruratKekerasanAparat, dan #KawalPutusanMK *trending* di X pada tanggal 21 Agustus 2024. Tanda pagar tersebut dicuitkan hingga sekitar 25900 kali, mengindikasikan fokus masyarakat terhadap fenomena ini. Fenomena yang terjadi

di Indonesia tersebut berkaitan dengan isu Dewan Perwakilan Rakyat (DPR) yang mengabaikan putusan Mahkamah Konstitusi (MK) terkait syarat pencalonan untuk Pemilihan Kepala Daerah (PILKADA) 2024. Fenomena ini menarik diteliti karena mengidentifikasi korelasi aspirasi mekanisme demokrasi terhadap berbagai kebijakan publik di Indonesia melalui media sosial. Oleh karena itu, diperlukan penerapan kemajuan teknologi untuk mendeteksi dan mengklasifikasi topik opini masyarakat pada saat dan setelah fenomena tersebut *trending*. Sehingga mendukung pemerintah dan pemangku kepentingan lainnya menentukan kebijakan yang tepat.

Klasifikasi data dalam bentuk teks semakin populer dengan penerapan model *Machine Learning* (ML) dan modifikasinya. Namun, arsitektur yang ada belum cukup efektif dalam mengidentifikasi faktor-faktor yang relevan sehingga akurasi kurang optimal. Mengacu terhadap tantangan ini, *Deep Learning* (DL) hadir sebagai solusi dengan arsitektur lebih optimal mampu menangkap pola kompleks pada data. *Deep Learning* mengintegrasi ekstraksi fitur secara otomatis dari teks, mengidentifikasi konteks, serta mengenali bahasa dengan lebih baik.

Penanganan *Natural Language Processing* (NLP) terdapat model DL optimal berbasis *Transformer*, perbandingan *Transformer* dengan model DL lainnya terdapat pada integrasi *self-attention*. *Self-attention* yang terdapat pada model *Transformer* merupakan integrasi memaknai hubungan kalimat atau paragraf dalam sebuah urutan otomatis tanpa memproses secara sekuensial (Vaswani dkk., 2017). Model *Bidirectional Encoder Representations from Transformers* (BERT) dikembangkan oleh Devlin dan lainnya pada tahun 2018. Model BERT merupakan model kontekstual performa unggul dalam berbagai tugas NLP. Model ini mempelajari konteks dari kiri ke kanan dan kanan ke kiri (Devlin dkk., 2019). Oleh karena itu, model BERT dapat menangkap konteks teks dengan lebih optimal.

Selama pengembangan model BERT yang terkenal adalah model *Indonesian Bidirectional Encoder Representations from Transformers* (IndoBERT), *Distilled Bidirectional Encoder Representations from Transformers* (DistilBERT), dan *Robustly Optimized Bidirectional Encoder Representations from Transformers Pretraining Approach* (RoBERTa). Model IndoBERT merupakan pengembangan

model BERT yang fokus pada analisis sentimen dan pengenalan entitas dalam teks bahasa Indonesia (Geni dkk., 2023). Model DistilBERT pengembangan lebih ringan dari model BERT yang dimodifikasi untuk mengoptimalkan ukuran model serta waktu inferensi tanpa mengorbankan akurasi (Sanh dkk., 2020). Model RoBERTa dikembangkan sebagai peningkatan model BERT dengan menawarkan optimasi lebih baik menggunakan *masking pattern* dan *dynamic masking* sehingga mampu menangkap konteks lebih panjang dan meningkatkan akurasi pada tugas-tugas NLP (Liu dkk., 2019). Selain itu, integrasi perkembangan telah mencapai model *Multilingual* yang mampu menangani berbagai bahasa dalam tugas NLP, seperti model DistilBERT *Multilingual* dan *Cross-lingual Language Model-Robustly Optimized Bidirectional Encoder Representations From Transformers Pretraining Approach* (XLM-RoBERTa).

Pada integrasinya, terdapat penelitian yang telah melakukan perbandingan terhadap model *Transformer* di antaranya model IndoBERT, DistilBERT *Multilingual*, dan XLM-RoBERTa untuk data Twitter dengan kategori informasi *adult* dan *non-adult* menghasilkan evaluasi optimal untuk model XLM-RoBERTa dengan *accuracy* 96% (Hidayatullah dkk., 2023). Namun, permasalahan yang ditemukan pada tugas NLP adalah data tidak terstruktur. Pada banyak kasus, model *Transformer* belum optimal mengatasinya karena cenderung memusatkan analisis pada keseluruhan konteks teks tanpa mengidentifikasi entitas secara tepat sehingga pola data tidak dapat dideteksi (Keraghel dkk., 2024). Hal tersebut dapat diatasi menggunakan *Named Entity Recognition* (NER). NER mampu mengurangi ambiguitas dengan mendeteksi entitas tertentu pada teks, seperti nama orang, organisasi, lokasi, tanggal, dan lain-lain. Selain itu, dalam identifikasi pola-pola tersembunyi menghasilkan sekumpulan topik dapat diatasi dengan *Latent Dirichlet Allocation* (LDA) (Huang, 2019). Sehingga implementasi NER dan LDA pada berbagai model *Transformer* tentunya meningkatkan tingkat akurasi dalam klasifikasi konteks isu kebijakan publik terkini di platform X.

Pengembangan penelitian dengan implementasi NER pada berbagai model *Transformer* telah banyak dilakukan dengan hasil yang baik. Penelitian yang dilakukan oleh Koto dkk., (2020), dengan tugas NER mencapai evaluasi optimal

menggunakan model IndoBERT dengan *F1-scores* 90%. Penelitian lain yang telah mengembangkan model BERT, DistilBERT, dan RoBERTa berbasis NER dilakukan oleh Berragan dkk., (2022), terkait ekstraksi nama tempat dari teks tak terstruktur bersumber dari Wikipedia memperoleh akurasi optimal untuk ketiga model *Transformer* sebesar 98%. Kemudian penelitian yang berfokus mengintegrasikan NER dan LDA pada data pertanian dilakukan oleh Gangadharan dan Gupta, (2020), menghasilkan analisis topik dari data kompleks mengadopsi kemampuan ekstrasi relasi dengan *human evaluation* menghasilkan akurasi 80%.

Berdasarkan uraian tersebut, peneliti akan melakukan perbandingan model *Transformer* untuk tugas klasifikasi teks menggunakan model IndoBERT, DistilBERT *Multilingual*, dan XLM-RoBERTa berbasis NER-LDA untuk deteksi entitas dan topik kebijakan publik. Tujuan memahami dinamika sosial dan politik terkini di Indonesia dalam bentuk aspirasi masyarakat di platform X. Pada penelitian ini, model yang dibandingkan mendapatkan perlakuan sama dalam menangani data tidak seimbang dengan parameter terbaik mencapai akurasi optimal.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Menganalisis keresahan masyarakat terhadap fenomena kebijakan publik yang *trending* pada platform X.
2. Mengoptimalkan proses klasifikasi *tweet* kebijakan publik pada data tidak terstruktur.
3. Menghasilkan perbandingan performa model *Transformer* berbasis NER-LDA dalam klasifikasi *tweet* kebijakan publik di platform X.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan perbandingan jumlah proporsi topik dari data saat dan setelah fenomena kebijakan publik *trending* pada platform X.
2. Mengintegrasikan pelabelan data dengan pendekatan NER-LDA untuk membangun model *Transformer,* yaitu model IndoBERT, DistilBERT *Multilingual*, dan XLM-RoBERTa dalam klasifikasi *tweet* kebijakan publik di platform X.
3. Mengidentifikasi parameter terbaik dalam membangun model *Transformer* pada klasifikasi *tweet* kebijakan publik di platform X berbasis NER-LDA menggunakan metrik evaluasi, yaitu akurasi, *precision, recall, F1-score*, serta ROC-AUC.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan gambaran komprehensif mengenai perubahan arah topik dan keresahan masyarakat terhadap fenomena kebijakan publik *trending* pada platform X.
2. Menangani ambiguitas makna pada data tidak terstruktur sebagai kontribusi pendekatan ekstraksi relasi tugas NLP.
3. Memberikan acuan pemerintah dan pemangku kepentingan mengoptimalkan kebijakan di Indonesia.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait sebagai integrasi rujukan dalam hasil analisis penelitian. Topik penelitian yang menjadi pembanding dan pembaharuan terdapat pada deteksi label menggunakan NER dan LDA berbasis model *Transformer* serta matriks evaluasi yang digunakan mulai dari *Accuracy* (*Acc*), *Precision* (*Prec*), *Recall* (*Rec*), hingga *F1-score* (*F1*). Secara mendalam terkait beberapa penelitian yang digunakan sebagai rujukan dengan rangkuman inti disajikan dalam Tabel 1.

Tabel 1. Penelitian Terkait Implementasi Model *Transformer* berbasis NER

No	Penelitian	Data	Metode	Hasil Terbaik (%)			
				Acc	***Prec***	***Rec***	***F1***
1	*Transformer Based Named Entity Recognition For Place Name Extraction From Unstructured Text* (Berragan dkk., 2022)	**Berita Britania Raya** Jumlah: 27334 Pelatihan: 21080 Validasi: 2907 Pengujian: 3347 Sumber: Abstrak Wikipedia Inggris	*Tagging*: NER Model: BERT, RoBERTa, DistilBERT	98	95	93	94
2	*Towards a Standardized Dataset on Indonesian Named Entity Recognition* (Khairunnisa dkk., 2020)	**Politik, sosial, dan ekonomi** Jumlah: 2340 Pelatihan: 1464 Validasi: 367 Pengujian: 509 Sumber: Kompas, Tempo, dan TribunNews.	*Tagging*: NER Model: IndoBERT, XLM-RoBERTa, mBERT	-	92	95	93

No	Penelitian	Data	Metode	Hasil Terbaik (%)			
				Acc	*Prec*	*Rec*	*F1*
3	*IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understandi-ng* (Wilie dkk., 2020)	**Domain bahasa Indonesia** Jumlah: 8400 Pelatihan:6720 Validasi: 840 Pengujian: 840 **Sumber: Berita**	*Tagging*: NER Model: IndoBERT, XLM-RoBERTa				79
4	*IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP* (Koto dkk., 2020)	**Domain bahasa Indonesia** Jumlah: 2125 Pelatihan: 1530 Validasi: 170 Pengujian: 425 Sumber: Berita	*Tagging*: NER Model: IndoBERT				90
5	*Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques* (Gangadharan dan Gupta, 2020)	**Pertanian** Jumlah: 3000 Sumber: Situs pertanian	*Tagging:* NER-LDA Model: *Conditional Random Fields* (CRF)	80			
6	*Adult Content Detection on Indonesian Tweets by Fine-tuning Transformer-based Models* (Hidayatullah dkk., 2023)	***Tweet* konten dewasa dan bukan** Jumlah: 54338 Pelatihan: 38036 Validasi: 5433 Pengujian: 10867 Sumber: Twitter	Klasifkasi Teks Model: IndoBERT, IndoBERTweet, mBERT, XLM-RoBERTa, DistilmBERT,	97	97	97	97

Berikut adalah ringkasan dari penelitian pada Tabel 1.

a. Penelitian yang dilakukan oleh Berragan dkk., (2022), model *Transformer* dibandingkan secara langsung dengan *Bidirectional Long Short-Term Memory* (BiLSTM) untuk tugas NER dengan fokus pada ekstraksi nama tempat dari teks yang tidak terstruktur. Data yang digunakan pada penelitian ini adalah abstrak Wikipedia Inggris yang berisi berita mengenai Britania Raya. Tahapan *text preprocessing* pada penelitian ini meliputi *regex cleaning, sentence segmentation, tokenization,* dan *annotation*.

Pada tahap *tagging*, digunakan skema BIOUL (*Beginning, Inside, Outside, Unit-length chunks,* dan *Last tokens*). Skema ini berfungsi untuk menandai nama tempat yang terdiri dari beberapa kata dengan lebih tepat. Contohnya, untuk nama tempat yang terdiri dari dua kata seperti "New York" kata pertama akan diberi label '*B-Place*' dan kata kedua '*L-Place*'. Pada nama tempat yang hanya terdiri dari satu kata akan diberi label *'U-Place'* dan kata yang bukan nama tempat akan diberi label 'O'. NER untuk pengenalan nama tempat dengan label ‘GPE’ (*Geopolitical Entity*), ‘LOC’ (*Location*), atau ‘FAC’ (*Facility*) menggunakan korpus nama tempat di Britania dari *GeoNames Gazetteer*.

Data yang digunakan dibagi menjadi tiga, yaitu data pelatihan, data validasi, dan data pengujian. Rasio pembagian sebesar 80:10:10, diperoleh data pelatihan dengan 21080 token berlabel, data validasi dengan 2907 token berlabel, serta data pengujian dengan 3347 token berlabel. Model *Transformer* yang dibangun dalam penelitian ini, yaitu model BERT, DistilBERT, dan RoBERTa. Seluruhnya menggunakan Adam *optimizer* untuk meningkatkan kinerja pelatihan. Perolehan evaluasi optimal pada ketiga model *Transformer* dibandingkan dengan model *Bidirectional Long Short-Term Memory* (BiLSTM). *Accuracy* untuk ketiga model *Transformer* mencapai 98% dengan hasil terbaik untuk model BERT dengan *precision* 95%, *recall* 93%, dan *F1-score* 94%.

b. Penelitian yang telah dilakukan Khairunnisa dkk., (2020), dalam analisisnya dengan tujuan mengatasi keterbatasan data yang tersedia serta meningkatkan akurasi konsistensi pengenalan entitas pada teks berbahasa Indonesia Penelitian ini menggunakan data topik politik, sosial, dan ekonomi yang diperoleh dari Kompas, Tempo, serta TribunNews. Tahapan pertama penghapusan tanda baca yang dilanjutkan dengan *tokenization*. Kemudian *tagging* NER untuk identifikasi entitas *location* (LOC), *organization* (ORG), and *person* (PER). Penelitian ini mengimplementasikan skema BIO (*Beginning, Inside, Outside*), sehingga setiap kata dalam kalimat ditandai sesuai dengan posisinya dalam entitas. Contohnya, kata pertama dari entitas diberi label 'B', kata di dalam entitas diberi label 'I', dan kata yang bukan bagian dari entitas diberi label 'O'.

Pada penelitian ini membandingkan pendekatan berbasis fitur (*feature-based*) dan pendekatan *fine-tuning* dengan berbagai model *Transformer.* Tahap awal data dibagi menjadi data pelatihan dan data pengujian dengan rasio 80:20. Kemudian 80% data pelatihan secara keseluruhan dibagi dengan data validasi menjadi rasio 80:20. Melalui pendekatan berbasis fitur dengan menggunakan model *Bidirectional Long Short-Term Memory - Conditional Random Field* (BiLSTM-CRF) yang diintegrasikan oleh beberapa *embedding*, termasuk *FastText*, mBERT, XLM-RoBERTa, dan IndoBERT. Parameter yang digunakan dalam BiLSTM-CRF, yaitu *learning rate* 0,1, *dropout* 0,5, *batch size* 32, dan maksimum 200 *epoch*. Kemudian untuk pendekatan *fine-tuning* dilakukan dengan *batch size* 32, jumlah *epochs* 5, dan *learning rate* ($3 \times 10^{-5}, 5 \times 10^{-5},$ dan 7×10^{-5}). Hasil metrik evaluasi menggunakan *precision*, *recall*, dan *F1-score* menunjukkan hasil yang baik untuk model IndoBERT. Pada penelitian ini menggunakan model BiLSTM-CRF diintegrasikan oleh *embedding* IndoBERT memperoleh *precision*, *recall*, dan *F1-score* sebesar 95%. Sedangkan untuk model *fine-tuning* IndoBERT dengan *precision* 92%, *recall* 95%, dan *F1-score* 93%.

c. Penelitian yang dilakukan oleh Wilie dkk., (2020), mencakup analisis 12 tugas NLP mulai dari *single sentence classification* hingga *pair-sentences sequence labeling* dengan berbagai level kompleksitas. Penelitian menggunakan data dengan berbagai gaya bahasa khusus domain bahasa Indonesia. Fokus pada integrasi model *pre-trained* IndoBERT menggunakan korpus Indo4B yang dikumpulkan dari teks sosia media, blog, berita, dan web.

   Fokus terhadap implementasi NER sebagai acuan penelitian. Data diperoleh dari *website* berita Indonesia terdiri dari 8000 kalimat. Implementasi NER menggunakan skema BIO (*Beginning, Inside, Outside*), terdiri dari 5 label PER (nama orang), LOC (nama lokasi), IND (nama produk), EVT (*event*), serta FNB (nama makanan dan minuman).

   Data yang digunakan dibagi menjadi tiga, yaitu data pelatihan, data validasi, dan data pengujian. Rasio pembagian sebesar 80:10:10, diperoleh data pelatihan 6720, data validasi 840, serta data pengujian 840. Penelitian fokus

mengintegrasikan model IndoBERT dengan hasil terbaik untuk NER diperoleh oleh model IndoBERT *large*, jumlah token maksimum 512, dan ukuran kosakata 30522. *Hyperparameter tuning* yang 12 digunakan, yaitu *batch size* [8,16], *learning rate* 4×10^{-5}, dan *dropout* 0,1. Perolehan evaluasi optimal dengan model IndoBERT *large*, yaitu *F1-scores* 79%

d. Penelitian yang dilakukan oleh Koto dkk., (2020), mencakup tugas *sequence labelling, semantic,* dan *coherency*. Penelitian menggunakan data dengan berbagai gaya bahasa khusus domain bahasa Indonesia. Fokus pada integrasi model *pre-trained* IndoBERT menggunakan korpus tiga sumber utama Wikipedia bahasa Indonesia (74 juta kata), artikel berita bahasa Indonesia dari Kompas, Tempo, dan Liputan6 (total 55 juta kata), serta web bahasa Indonesia (90 juta kata).

   Fokus terhadap implementasi NER sebagai acuan penelitian. Implementasi NER terdiri dari 2125 kalimat dengan tiga label, yaitu *person, location*, dan *organization*. Data melalui pembagian awal dengan rasio 80:20 untuk data pelatihan dan data pengujian. Kemudian pembagian kedua, untuk 80% dari data pelatihan dibagi menjadi data pelatihan dan data validasi dengan rasio 90:10.

   Penelitian memperoleh hasil terbaik oleh model *pre-trained* IndoBERT dengan 220 juta kata, fungsi aktivasi GELU, dan *optimizer* Adam. *Hyperparameter tuning* yang digunakan, yaitu *learning rate* 1×10^{-4} dan jumlah *epoch* 100. Kemudian menggunakan IndoBERT *embedding,* yaitu menambahkan token khusus di awal [CLS] dan di akhir [SEP]. Perolehan evaluasi optimal dengan model IndoBERT mencapai *F1-scores* 90%.

e. Penelitian yang dilakukan oleh Gangadharan dan Gupta, (2020), menganalisis NER untuk tugas yang lebih luas dengan integrasi LDA dalam identifikasi dokumen-dokumen pertanian. Data yang digunakan berasal dari situs pertanian dengan total 3000 kalimat. Pada penelitian ini, NER diterapkan menggunakan *Conditional Random Field* (CRF) untuk mengekstrak entitas dari teks. Korpus yang digunakan adalah *Agricultural Vocabulary* (AGROVOC), berisi 36000 kata dalam 33 bahasa, yang memberikan cakupan luas untuk entitas pertanian.

Namun, karena label entitas yang lebih spesifik diperlukan untuk analisis yang lebih mendalam, setiap kata benda (*noun*) pada POS *Tagging* diintegrasikan dengan NER-LDA untuk mengidentifikasi topik-topik yang relevan, yaitu *disease*, *soil*, dan *fertilizer*.

Interpretasi menggunakan *pyLDAvis* sebagai alat visualisasi terhadap ketiga topik tersebut dengan *human evaluation* akurasi mencapai 80%. Penelitian ini menunjukkan bahwa penggabungan teknik NER dan LDA dapat memberikan wawasan yang lebih tajam dan lebih terstruktur mengenai topik teks pertanian dengan membantu dalam proses ekstraksi relasi.

f. Penelitian yang dilakukan oleh Hidayatullah dkk., (2023), bertujuan mengklasifikasikan konten dewasa dalam teks pada media sosial Twitter berbahasa Indonesia. Pada penelitian digunakan data Twitter yang terdiri dari 54338 *tweet*, mencakup dua kategori utama konten dewasa sebanyak 27146 *tweet* dan bukan dewasa 27192 *tweet*. Penelitian ini mengimplementasikan lima model Transformer, yaitu model IndoBERT, IndoBERTweet, mBERT, XLM-RoBERTa, dan DistilmBERT. Data dibagi menjadi tiga bagian, yaitu 70% untuk pelatihan, 10% untuk validasi, dan 20% untuk pengujian.

   Pada integrasi meningkatkan performa model, dilakukan *hyperparameter tuning* dengan pendekatan *grid search* menggunakan *Optuna*. Parameter yang disesuaikan meliputi *learning rate* ($1 \times 10^{-4}, 3 \times 10^{-4}, 2 \times 10^{-5}, 3 \times 10^{-5}$, dan 5×10^{-5}), *batch size* (8, 16, 32, 64, 128), serta *weight decay* (0.001, 0.01, 0.1). Proses pelatihan dilakukan selama lima *epoch* menggunakan Adam *optimizer*. Hasil penelitian menunjukkan bahwa semua model *Transformer* yang diuji mampu mengklasifikasikan *tweet* secara optimal dengan model XLM-RoBERTa memberikan performa kinerja terbaik mencapai *accuracy* sebesar 97%.

2.2 *Natural Language Processing*

Natural Language Processing merupakan cabang dari kecerdasan buatan dan linguistik komputasional dengan fokus integrasi antara komputer serta bahasa alami

manusia sehingga menjadi lebih terstruktur (Fanni dkk., 2023). Pada NLP, mesin dilatih untuk memahami makna bahasa serta memberikan respons yang sesuai dengan konteks. Proses ini melibatkan berbagai langkah, termasuk integrasi representasi numerik yang dapat diolah oleh mesin memungkinkan untuk belajar dan memahami dunia. Tahap penting dalam NLP adalah *text preprocessing* yang dapat meningkatkan akurasi pada klasifikasi teks, secara umum langkah awalnya sebagai berikut (Yuyun dkk., 2023):

1. *Case folding*: tahapan mengonversi seluruh huruf kapital menjadi huruf kecil, sehingga dapat mengintegrasi standar format teks.
2. *Cleaning*: tahapan menghilangkan karakter yang bukan alfanumerik, seperti spasi, emoji, simbol, serta tanda baca.
3. *Filtering* atau *stopword removal*: kata-kata umum yang sering muncul dan tidak memiliki makna signifikan dikenal sebagai *stopword*. Termasuk kata penghubung seperti "dan", "yang", "serta", "setelah", dan sebagainya. Menghilangkan *stopword* dapat mengurangi ukuran indeks, mempercepat waktu pemrosesan, serta menurunkan tingkat kebisingan dalam data.
4. *Tokenization*: tahapan memecah atau membagi teks menjadi unit-unit yang lebih kecil sehingga dapat disebut token.
5. *Stemming*: tahapan mengubah kata ke bentuk dasarnya sehingga mencari *root* dari setiap kata. Mampu mengurangi jumlah indeks yang berbeda dalam suatu data sehingga kata yang memiliki sufiks atau prefiks dapat diintegrasikan ke bentuk asalnya. *Stemming* dapat meningkatkan *recall*, tetapi dapat mengurangi *precision* karena kata-kata dengan makna yang berbeda mendapat kesempatan menjadi bentuk yang sama. Hal tersebut berdampak signifikan pada *preprocessing*. Oleh karena itu, digunakan *multi-phase stemmer* yang dimodifikasi berfungsi dengan baik pada kata baku dan tidak baku (Boban dkk., 2020).

Text preprocessing bertujuan mengubah teks yang tidak terstruktur terintegrasi sebagai data yang lebih terstruktur, sehingga mesin dapat melakukan pemrosesan teks komprehensif (Kumar dkk., 2020).

2.3 *Text Mining*

Text Mining merupakan mekanisme identifikasi korelasi pada informasi yang mengintegrasikan interaksi antara pengguna serta sekumpulan dokumen menggunakan alat analisis yang merupakan bagian dari *data mining* (Qamar dan Raza, 2020). Menurut Kumar dkk., (2020) *text mining* menerapkan disiplin ilmu berfokus pada teknik otomatis untuk mengekstrak informasi dan mengidentifikasi pola signifikan yang tersembunyi dari dokumen atau data tidak terstruktur. *Text mining* mengintegrasikan berbagai disiplin ilmu, seperti linguistik, statistik, serta ilmu komputer. Perkembangan saat ini menunjukkan pertumbuhan data yang besar, sehingga *text mining* fokus sebagai alat penting dalam mengelola serta menganalisis data teks secara komprehensif untuk menemukan wawasan baru dalam berbagai bidang. Implementasi *text mining* umumnya dalam analisis artikel, situs web, blog, platform media sosial, dan lain sebagainya. Namun, tantangan yang ada pada *text mining* ialah konteks pembicaraan yang tidak selalu jelas. Menurut Hassani dkk., (2020) tantangan tersebut hadir pada teks yang berhubungan dengan kebijakan publik. Oleh karena itu, perlu dibangun model klasifikasi optimal dalam memahaminya "*knowledge of what-goes-with-what*" (Alexandre dkk., 2024).

2.4 Klasifikasi Teks

Klasifikasi teks adalah proses pengelompokan data teks ke dalam kategori atau label tertentu berdasarkan makna yang terkandung dalam teks tersebut (Cunha dkk., 2023). Klasifikasi teks termasuk *supervised learning* sebagai pendekatan pembelajaran mesin dengan fokus menentukan label menggunakan data yang telah diberi label sebelumnya. Menurut Shah dkk., (2020) klasifikasi teks secara matematis dapat dinyatakan sebagai pemetaan yang memiliki fungsi ditunjukkan pada Persamaan (1):

$$f: X \rightarrow Y \tag{1}$$

Keterangan:

X = ruang fitur (vektor representasi teks).

Y = ruang label (kategori atau kelas).

Fungsi keputusan bertujuan memisahkan kelas-kelas yang berbeda berdasarkan informasi fitur atau kriteria tertentu pada input. Dengan demikian, klasifikasi teks mengintegrasikan penerapan algoritma yang dirancang mempelajari pola data pelatihan dan menerapkannya pada data baru untuk menentukan label yang sesuai. Klasifikasi teks dapat dibedakan berdasarkan jumlah label atau kategori yang tersedia, yaitu klasifikasi biner dan klasifikasi *multiclass* (Mohammed dkk., 2020). Perbedaan pada kedua jenis klasifikasi tersebut diilustrasikan dalam Gambar 1.

Binary-class Dataset

X_1	X_2	…	X_m	Y
X_{11}	X_{12}	…	X_{1m}	1
X_{21}	X_{22}	…	X_{2m}	0
…	…	…	…	1
X_{n1}	X_{n2}	…	X_{nm}	0

Multi-class Dataset

X_1	X_2	…	X_m	Y
X_{11}	X_{12}	…	X_{1m}	C1
X_{21}	X_{22}	…	X_{2m}	C2
…	…	…	…	C3
X_{n1}	X_{n2}	…	X_{nm}	C4

Gambar 1. Jenis Klasifikasi Teks Berdasarkan Jumlah Kategori (Mohammed dkk., 2020).

Berdasarkan Gambar 1, klasifikasi biner ialah klasifikasi teks yang menyertakan dua label atau kategori, sehingga setiap data hanya dapat dikategorikan sebagai salah satu dari dua peluang kemungkinan. Klasifikasi *multiclass* mencakup lebih dari dua kelas, sehingga data dapat dikategorikan ke dalam berbagai label yang berbeda. Selain itu, klasifikasi teks juga dapat dibagi berdasarkan hubungan antar kategori, yaitu klasifikasi ordinal serta klasifikasi nominal (Li dkk., 2022). Klasifikasi ordinal melibatkan kategori yang memiliki urutan atau hierarki tertentu, sehingga setiap kelas atau label menunjukkan tingkat yang berbeda. Sedangkan klasifikasi nominal melibatkan kategori yang tidak memiliki urutan atau hierarki, sehingga setiap kelas setara.

Label atau kategori pada klasifikasi nominal, yaitu kategori konteks dari isu kebijakan publik. Ruang lingkup dari kebijakan publik sangat luas karena mencakup berbagai bidang, yaitu politik, ideologi, pertahanan dan keamanan, ekonomi, sosial budaya, pendidikan, kesehatan, demografi, dan sumber daya alam (Alim, 2024). Berdasarkan pemahaman terhadap pengklasifikasian kategori ruang lingkup tersebut, analisis dapat lebih efektif dilakukan secara mendalam sehingga mencapai tujuan yang diinginkan.

2.5 *Crawling*

Crawling adalah teknik yang digunakan untuk mengekstrak data dari berbagai sumber di web menggunakan bantuan *software* komputer dengan bahasa pemrograman (Khder, 2021). Tanpa melakukan *copy-paste* informasi secara manual dari situs web, web *crawling* melalui komputer dapat melakukannya dengan lebih akurat serta jauh lebih cepat dibandingkan manusia. *Crawling* bukan hanya tentang pengambilan data, tetapi juga mempertimbangkan cara penggunaan data secara bertanggung jawab serta membangun dampak positif.
Web *crawling* telah banyak dilakukan untuk memperoleh data target analisis. Salah satunya web *crawling* menggunakan token Twitter atau yang saat ini dikenal dengan X.

Hal tersebut merupakan metode efektif untuk mengekstrak data dari platform media sosial dengan cara terstruktur serta mematuhi kebijakan yang ada. Token API terdiri dari beberapa elemen, seperti *Consumer Key*, *Consumer Secret, Access Token*, dan *Access Token Secret*. Secara keseluruhan token tersebut sebagai mekanisme autentikasi yang memastikan bahwa akses ke data X hanya diberikan kepada pengguna yang terotorisasi. Metode ini tidak hanya memberikan akses data yang lebih besar secara *real-time*, tetapi juga menghindari pemblokiran secara langsung. *Crawling* bekerja secara otomatis, informasi yang dikumpulkan berdasarkan pada kata kunci yang disediakan oleh pengguna. Mematuhi kebijakan penggunaan Twitter API, teknik ini mampu memperoleh wawasan berharga dari data yang relevan, termasuk analisis konteks terhadap kebijakan publik yang menjadi permasalahan masyarakat luas.

2.6 *Named Entity Recognition*

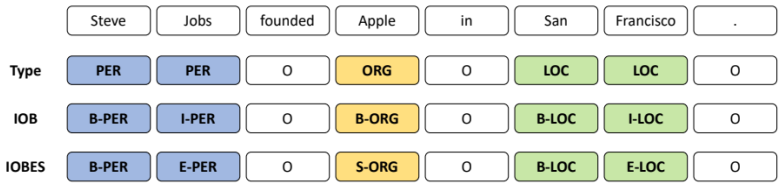
Langkah pertama yang krusial dalam mengekstraksi fakta relasional antar entitas adalah mengidentifikasi bagaimana dan di mana entitas-entitas tersebut disebutkan dalam teks. Sebelum melakukan ekstraksi hubungan antar entitas, hal yang harus

diperhatikan adalah bagaimana entitas-entitas tersebut dapat dikenali dengan tepat dalam teks (Taille dkk., 2022).

Named Entity Recognition (NER) adalah tugas dalam NLP yang berfokus dalam mengidentifikasi atau mengkategorikan entitas tertentu dari teks (*named entities)* (Khairunnisa dkk., 2023). Kategori tersebut telah ditentukan sebelumnya, seperti nama orang, lokasi, organisasi, tanggal, waktu, nominal uang, persentase, serta *miscellaneous entities.* NER berfungsi sebagai komponen awal *Information Extraction* (IE) serta penghubung penting antara data teks tidak terstruktur dengan informasi yang terstruktur (Wilie dkk., 2020). Pengembangan *tagging* pada *text chunking* dengan skema IOB *tagging* pertama kali diperkenalkan oleh Ramshaw dan Marcus, (1995). Skema ini mengindikasikan posisi kata dalam sebuah entitas yang lebih besar, dengan representasi sebagai berikut:

- B untuk kata pertama dalam entitas (*Beginning*).
- I untuk kata yang berada di dalam entitas (*Inside*).
- O untuk kata yang tidak termasuk dalam entitas (*Outside*).

Berdasarkan representasi tersebut, contoh penerapan skema IOB *tagging* dalam NER dapat dilihat seperti pada Gambar 2.



Gambar 2. Skema IOB *Tagging* NER (Taille dkk., 2022).

Representasi Gambar 2 untuk skema IOB *tagging* dalam NER diperluas dengan skema IOBES atau BILOU. Pada IOBES tambahan *tag* E untuk kata terakhir dalam entitas (*End*) dan S untuk entitas yang hanya terdiri dari satu kata (*Singleton*). Pada skema BILOU, tag E digantikan dengan L (*Last*) dan tag S digantikan dengan U (*Unit*). Melalui kemajuan dalam *deep learning*, NER telah mencapai tingkat akurasi yang tinggi dan menjadi alat penting untuk deteksi entitas dalam berbagai konteks kalimat. Pada banyak penelitian implementasi NER telah mengadopsi arsitektur *Transformer*. Salah satunya arsitektur *BertForTokenClassification* yang telah

terintegrasi dengan pembelajaran bahasa Indonesia mengadopsi model BERT yang terdiri dari tumpukan *encoder* arsitektur *Transformer*. Terdiri dari 39 entitas yang diuraikan pada Tabel 2.

Tabel 2. Kategori NER

ID	Entitas
0	B-CRD
1	B-DAT
2	B-EVT
3	B-FAC
4	B-GPE
5	B-LAN
6	B-LAW
7	B-LOC
8	B-MON
9	B-NOR
10	B-ORD
11	B-ORG
12	B-PER
13	B-PRC
14	B-PRD
15	B-QTY
16	B-REG
17	B-TIM
18	B-WOA
19	I-CRD
20	I-DAT
21	I-EVT
22	I-FAC
23	I-GPE
24	I-LAN

ID	Entitas
25	I-LAW
26	I-LOC
27	I-MON
28	I-NOR
29	I-ORD
30	I-ORG
31	I-PER
32	I-PRC
33	I-PRD
34	I-QTY
35	I-REG
36	I-TIM
37	I-WOA
38	O

Berdasarkan Tabel 2, arsitektur memiliki maksimal *position embedding* 512 token, 12 lapisan *Transformer*, 12 *attention head*. *Pre-trained* menggunakan korpus 220 juta kata dengan sumber Wikipedia, Kompas, Tempo, Liputan6, dan *Indonesian Web Corpus.*

2.7 *Latent Dirichlet Allocation*

Latent Dirichlet Allocation (LDA) adalah *generative probabilistic* model yang digunakan untuk mengidentifikasi topik-topik semantik tersembunyi dalam kumpulan data teks. Intuisi dasarnya mendeteksi sebuah dokumen dapat mengandung berbagai topik. LDA dapat digunakan untuk meringkas, mengelompokkan, menghubungkan, atau memproses data yang sangat besar karena

LDA menghasilkan daftar topik yang diberi bobot untuk setiap dokumen (Campbell dkk., 2015). Penggunaan LDA dijelaskan melalui representasi matematis yang dapat dilihat dalam Persamaan (2) (Setijohatmo dkk., 2020).

$$p(w, z, \vartheta, \varphi|\alpha, \beta) = p(\varphi|\beta)p(\theta\ |\alpha)p(z, \theta\ )p(w|\varphi k) \quad (2)$$

Keterangan:

β = *dirichlet* parameter atas distribusi kata terhadap topik.

φ = distribusi kata terhadap topik dalam *corpus*.

K = kumpulan topik.

W = kata.

N = kumpulan kata.

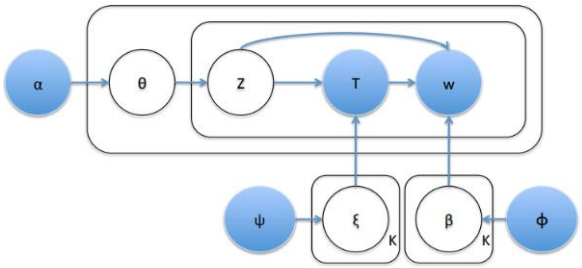
M = kumpulan dokumen.

Z = topik index assignment.

θ = dokumen.

α = *dirichlet* parameter atas distribusi topik terhadap dokumen.

Pada perkembangan integrasinya, terdapat cara kerja LDA yang menggabungkan pendekatan NER bertujuan mendeteksi informasi entitas untuk menghasilkan topik-topik lebih informatif dan relevan (Gangadharan dan Gupta, 2020). Proses generatif NER-LDA ditunjukkan dalam representasi Gambar 3.



Gambar 3. Representasi pendekatan NER-LDA (Nallapati dkk., 2010).

Representasi Gambar 3 untuk pendekatan NER-LDA mengacu pada proses sebagai berikut:

1. Pada setiap topik $k\ \in 1, \ldots, K$:
   - Menghasilkan distribusi multinomial atas label NER untuk topik k: $\xi_k \sim Dir(\cdot\ |\Psi)$.
2. Pada setiap label NER t:

- Menghasilkan distribusi multinomial untuk kata-kata berdasarkan topik k atas label NER untuk topik t: $\beta_{k,t} \sim Dir(\cdot|\Phi)$.

3. Pada setiap dokumen $d \in 1, \ldots, M$:
   - Menghasilkan distribusi dokumen atas topik $\theta_d \sim Mult(\cdot|\alpha)$.
4. Pada setiap posisi $n \in 1, \ldots, N_d$:
   - Menghasilkan indikator topik $z_{dn} \sim Mult(\cdot|\theta_d)$.
   - Menghasilkan label NER $t_{dn} \sim Mult\left(\cdot \middle| \xi_{z_{dn}}\right)$.
   - Menghasilkan kata $w_{dn} \sim Mult\left(\cdot \middle| \beta_{z_{dn},t_{dn}}\right)$.

Metrik yang digunakan untuk mengevaluasi kualitas topik yang dihasilkan oleh LDA adalah *coherence score*. Metrik ini mengukur sejauh mana kata-kata dalam suatu topik memiliki keterkaitan yang tinggi, mencerminkan interpretabilitas topik tersebut. Salah satu pendekatan yang umum digunakan adalah metode C_V atau *coherence value*, mengombinasikan *Normalized Pointwise Mutual Information* (NPMI) dan *cosine similarity* untuk menilai kesamaan antara pasangan kata dalam suatu topik. Nilai *coherence* akhir diperoleh dengan menghitung rata-rata skor *coherence* dari semua topik yang dihasilkan oleh model. *Coherence* metode C_V direpresentasikan pada Persamaan (3) hingga (5) (Syed dan Spruit, 2017).

$$NPMI(w_i, w_j) = \frac{\log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j}}{-\log P(w_i, w_j) + \epsilon} \tag{3}$$

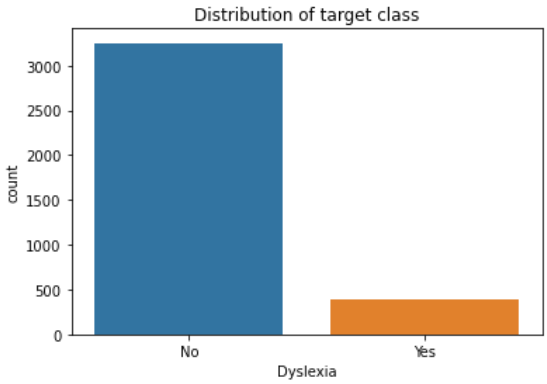
$$Cosine = \frac{u \cdot w}{\|\vec{u}\|_2 \cdot \|\vec{w}\|_2} \tag{4}$$

$$C_V = \frac{1}{K} \sum_{m=0}^{K-1} \sum_{i \neq j} Cosine\left(NPMI(w_i, w_j)\right) \tag{5}$$

2.8 Ketidakseimbangan Data

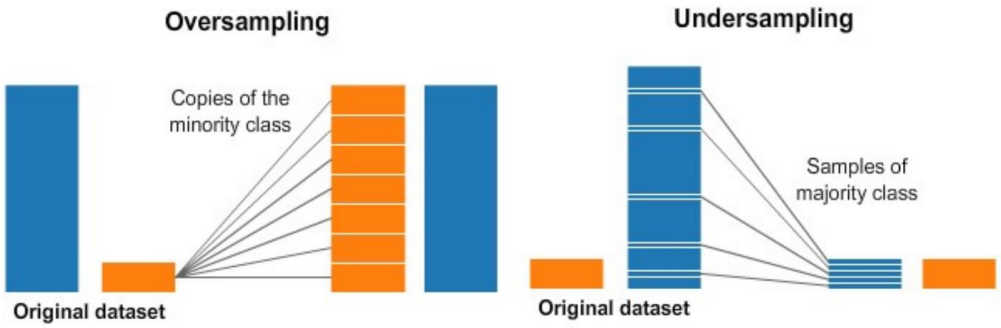
Ketidakseimbangan kelas data terjadi ketika distribusi jumlah sampel dalam setiap kelas berbeda secara signifikan, dengan satu kelas memiliki lebih banyak sampel daripada kelas lainnya. Data yang tidak seimbang ini menyebabkan model lebih sulit mengenali kelas minoritas sehingga menghasilkan akurasi yang rendah untuk kelas tersebut (Syukron dan Subekti, 2018). Meskipun kesalahan klasifikasi dapat

dikurangi dengan mengklasifikasikan semua contoh ke dalam kelas mayoritas, hal ini berisiko membuat kelas minoritas lebih rentan mengalami kesalahan klasifikasi. Distribusi kelas data yang tidak seimbang diilustrasikan pada Gambar 4.



Gambar 4. Ketidakseimbangan Data (Syukron dan Subekti, 2018).

Berdasarkan Gambar 4, terdapat tiga pendekatan dalam mengatasinya, yaitu *oversampling*, *undersampling*, serta pendekatan *hybrid* mengintegrasikan keduanya yang diilustrasikan pada Gambar 5.



Gambar 5. Ilustrasi *Oversampling* dan *Undersampling* (Syukron dan Subekti, 2018).

Berdasarkan Gambar 5, *oversampling* diintegrasikan dengan pendekatan menambah jumlah data pada kelompok minoritas untuk mencapai keseimbangan jumlah data kelompok mayoritas. Sebaliknya, *undersampling* merupakan pendekatan dalam mengurangi jumlah data kelompok mayoritas yang kurang relevan sehingga mencapai keseimbangan jumlah data dengan kelompok minoritas. Selain itu, pendekatan *hybrid* mengintegrasikan tenik *oversampling* dan *undersampling* disesuaikan dengan kebutuhan spesifik dari karakteristik kumpulan data yang dimiliki.

2.9 Augmentasi Data Teks

Augmentasi data teks adalah proses memodifikasi data teks untuk meningkatkan variasi dalam *dataset* tanpa mengubah maknanya. Pendekatan bertujuan untuk meningkatkan kualitas pelatihan model, terutama dalam masalah data terbatas atau tidak seimbang. Pendekatan sederhana seperti *resampling* umumnya tidak efektif dapat menyebabkan *overfitting* dan *overgeneralization* sehingga menghasilkan sampel yang tidak informatif (Shorten dkk., 2021). Oleh karena itu, untuk menciptakan *dataset* yang lebih kaya dan beragam, augmentasi data teks hadir sebagai pendekatan lebih *robust* terhadap variasi teks dengan menangani berbagai pola bahasa. Hal ini penting untuk meningkatkan generalisasi model, khususnya pada tugas-tugas NLP. Jenis-jenis augmentasi data teks sebagai berikut:

a. *Easy Data Augmentation* (EDA)

Easy Data Augmentation mencakup beberapa teknik yang efektif untuk meningkatkan variasi teks diilustrasikan pada Tabel 3.

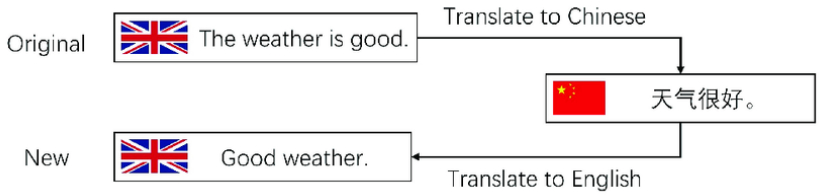
Tabel 3. Penerapan *Easy Data Augmentation*

Easy Data Augmentation	**Contoh**
Random Swapping	*I am jogging → I tiger jogging*
Random Insertion	*I am jogging → I am salad jogging*
Random Deletion	*I am jogging → I jogging*
Synonym Replacement	*I am jogging → I am running*

- *Random Swapping*: Menukar posisi dua kata secara acak untuk memperkenalkan perbedaan struktur kalimat.
- *Random Insertion*: Menambahkan kata secara acak ke dalam teks untuk memperkenalkan variasi pada struktur kalimat.
- *Random Deletion*: Menghapus beberapa kata secara acak membuat kalimat lebih pendek sehingga melatih model memahami konteks dengan informasi lebih sedikit.
- *Synonym Replacement*: Mengganti beberapa kata dalam teks dengan sinonimnya untuk menciptakan variasi tanpa mengubah makna.

b. *Backtranslation*

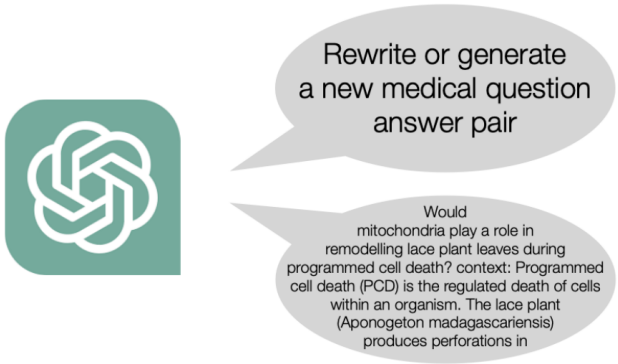
Teknik ini melibatkan penerjemahan teks ke bahasa lain dan kembali lagi ke bahasa asal. Proses ini menghasilkan variasi teks yang tetap bermakna, tetapi memiliki struktur kalimat berbeda. Integrasi teknik ini banyak dilakukan dengan efisiensi memperkaya data pelatihan lebih beragam secara linguistik sehingga dapat membantu model memahami konteks lebih luas. Teknik *backtranslation* dapat diilustrasi dengan contoh pada Gambar 6.



Gambar 6. *Backtranslation* (Wang dkk., 2023).

c. *Generative Models*

Generative models digunakan untuk membuat teks baru yang menyerupai data asli. Pendekatan mempelajari pola dari data, model dapat menghasilkan teks sintetis yang kaya variasi dan tetap relevan dengan konteks. Teknik ini sangat efektif untuk menangani masalah data tidak seimbang, terutama untuk kelas yang kurang terwakili. *Generative model*s beberapa diantaranya yaitu GPT, Gemini, dan LLaMA dengan contoh ilustrasi penerapannya pada Gambar 7.



Gambar 7. *Generative Data Augmentation* (Guo dkk., 2023).

2.10 *Word Embedding*

Perkembangan integrasi dalam pemrosesan teks atau NLP terus meningkat, salah satunya pengembangan bahasa dalam mengonversi kata-kata menjadi vektor

dengan menghubungkan pemahaman manusia serta pembelajaran mesin. Integrasi ini bertujuan dalam merepresentasikan kata dengan lebih baik dalam ruang n-dimensi sehingga akurasi dan performa model lebih optimal (Birunda dan Devi, 2021). Namun, dalam implementasinya terdapat berbagai tantangan termasuk dalam tugas NLP yang membutuhkan integrasi fitur dalam mengatasi tidak strukturnya data.

Menurut Nurdin dkk., (2020), *word embedding* memetakan setiap kata pada kalimat ke dalam dimensi vektor, sehinga implementasi vektor mempresentasikan proyeksi kata di dalam ruang vektor. Posisi kata tersebut dipelajari dari teks atau berdasarkan kata-kata di sekitarnya. *Word embedding* ini dapat menangkap makna semantik dan sintaktik kata yang dilatih dengan korpus. *Word embedding* dapat dibuat langsung dari kumpulan data yang dimiliki atau menggunakan *pre-trained word embedding* yang telah tersedia. *Pre-trained word embedding* merupakan *word embedding* yang telah dilatih menggunakan kumpulan data yang besar pada domain permasalahan tertentu yang dapat digunakan untuk menyelesaikan permasalahan lain yang serupa. Penggunaan *word embedding* ini harus disesuaikan dengan data yang dimiliki (Nurdin dkk., 2020). Jenis-jenis *word embedding* umumnya, yaitu tradisional *word embedding, static word embedding,* serta *contextualized word embedding.*

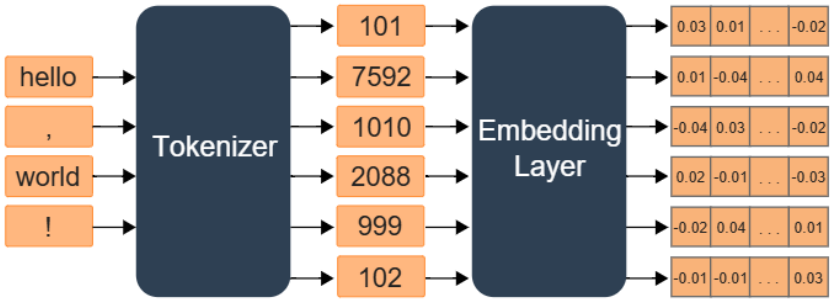
Pada tahun 2018, tim peneliti dari Google AI *Language* memperkenalkan implementasi model BERT. Pendekatan *Transformer* diadopsi dalam model BERT dengan integrasi mekanisme untuk memahami hubungan kontekstual antar kata-kata dalam teks melalui *self-attention mechanism* (Tanaka dkk., 2020). Pada *self-attention mechanism,* setiap kata dalam teks dapat saling berinteraksi dan menentukan seberapa besar perhatian yang harus diberikan kepada kata-kata lainnya. Pada BERT *embedding* representasi urutan kata dalam suatu kalimat dihasilkan dengan mengaitkan kata-kata yang berbeda dalam urutan yang sama menggunaan mekanisme *encoder.* Hal ini memungkinkan pemahaman yang lebih baik tentang konteks dan hubungan antara kata-kata dalam teks.

Pada BERT *embedding* merepresentasikan data teks dengan *token embeddings* W_N, *segmentation embeddings* S_N, dan *position embeddings* P_N. Formula BERT

embedding direpresentasikan pada Persamaan (6) (Nandanwar dan Choudhary, 2023).

$$E_N = W_N + S_N + P_N \quad (6)$$

Token *embeddings* diperoleh dengan mengonversi teks menjadi daftar token ID menggunakan *tokenizer* BERT. Setiap token ID memiliki *embedding* unik yang telah dipelajari selama pelatihan model sebagai contoh dapat dilihat pada Gambar 8.

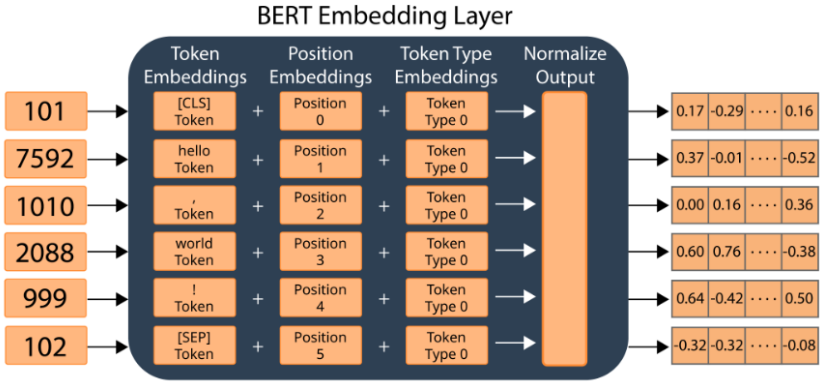


Gambar 8. Token *Embeddings* (Nandanwar dan Choudhary, 2023).

Berdasarkan Gambar 8, integrasi dilanjutkan pada *token type embeddings* atau *segment embeddings*. Integrasi tersebut digunakan untuk membedakan antara dua segmen teks dalam tugas *Next Sentence Prediction* (NSP). Kemudian, penerapan *position embeddings* untuk mengintegrasikan posisi token dalam suatu urutan. Penerapan tersebut dilakukan sebagai integrasi representasi posisi kata dalam kalimat secara eksplisit. *Position embeddings* memiliki ukuran tetap sesuai dengan panjang maksimum input yang diterima oleh model. Sebagai contoh ilustrasi penerapan *position embeddings* disajikan pada Gambar 9.

Gambar 9. *Position Embeddings* (Nandanwar dan Choudhary, 2023).

Berdasarkan Gambar 9, diperoleh penerapan BERT *Embedding Layer* bertanggung jawab untuk menggabungkan ketiga *embedding* menjadi representasi akhir untuk setiap token. Proses ini dilakukan dengan menjumlahkan *token embeddings*, *token type embeddings* (*segment embeddings)*, dan *position embeddings* lalu menerapkan normalisasi. Ilustrasi proses tersebut disajikan pada Gambar 10.



Gambar 10. Ilustrasi BERT *Embedding Layer* (Nandanwar dan Choudhary, 2023).

Berdasarkan Gambar 10, *output* dari *layer* memiliki tiga dimensi yang diilustrasikan pada Gambar 11.

Gambar 11. Dimensi *Output Layer* (Nandanwar dan Choudhary, 2023).

Berdasarkan Gambar 11, dimensi terdiri dari *batch size*, jumlah token, dan *hidden size.* Dengan demikian, *embedding* akhir dari suatu kata mencerminkan informasi semantik dari kata tersebut, segmen, dan posisinya dalam kalimat.

2.11 *Splitting* Data

Splitting data merupakan integrasi yang bertujuan dalam membagi data menjadi beberapa bagian untuk keperluan pelatihan dan evaluasi pembelajaran model. Pembagian ini bertujuan menghilangkan atau mengurangi bias pada data pelatihan selama pembelajaran model dengan mampu generalisasi data baru yang belum pernah dilihat sebelumnya. Implementasi berbagai rasio bertujuan mencegah algoritma menghasilkan *overfitting* yang dapat berperforma buruk pada data pengujian sebenarnya (Kahloot dan Ekler, 2021).

Salah satu metode yang umum digunakan dalam pengembangan model adalah membagi data menjadi tiga bagian utama, yaitu *training*, *validation*, dan *testing*. *Training* data digunakan untuk melatih model agar dapat memahami pola dalam data dengan menyesuaikan parameter model. *Validation* data membantu dalam

proses pemilihan model dengan mengevaluasi kinerja model terhadap data yang tidak digunakan selama pelatihan, sehingga dapat mencegah *overfitting*. Kemudian *testing* data digunakan mengevaluasi akurasi dan keandalan model pada data baru. Menurut Saunshi dkk., (2021), pembagian ini penting untuk memastikan bahwa model tidak hanya belajar dari data pelatihan, tetapi juga mampu memberikan hasil yang baik pada data yang belum pernah dilihat sebelumnya. Tingkat kesalahan yang dihitung dari data pengujian memberikan seberapa baik model dapat belajar data di dunia nyata.

2.12 Fungsi Aktivasi

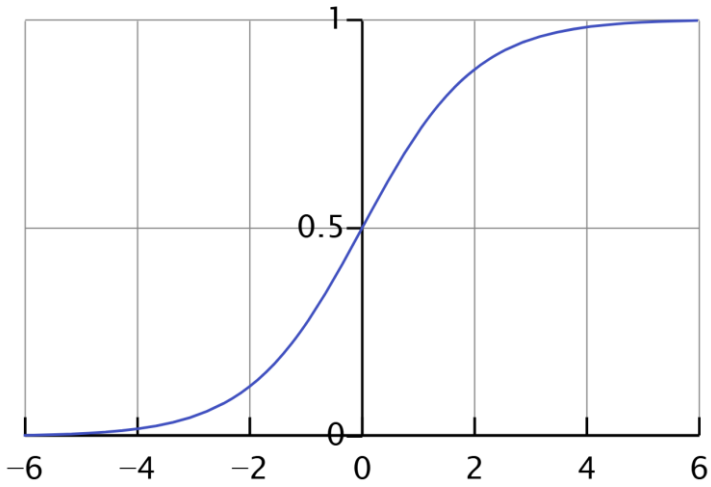
Menurut Dubey dkk., (2022), fungsi aktivasi merupakan mekanisme yang mengubah nilai *input* menjadi *output* yang kemudian diteruskan ke lapisan berikutnya sehingga meningkatkan kemampuan model dalam menangkap pola kompleks dalam data. Akurasi prediksi dari suatu model sangat bergantung pada jumlah lapisan serta jenis fungsi aktivasi yang digunakan dengan menangani non linieritas. Fungsi aktivasi mampu merepresentasikan pola *non-linear* dengan lebih efisien. Secara bertahap menyusun representasi yang lebih kompleks dari setiap token dalam teks dengan mempertahankan makna sesuai konteks. Fungsi aktivasi yang umum digunakan antara lain *sigmoid*, *softmax*, *Tangen Hiperbolik* (tanh), *Rectified Linier Unit* (ReLU), dan *Gausssian Error Linier Unit* (GELU).

a. *Sigmoid*

   Fungsi aktivasi *sigmoid* merupakan fungsi *non-linear* yang terdiferensiasi terbatas serta didefinisikan untuk setiap nilai *input* nyata memiliki turunan bernilai positif. Representasi matematika untuk fungsi aktivasi *sigmoid* terdapat pada Persamaan (7) (Younisse dkk., 2023).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

   Fungsi tersebut memetakan setiap nilai *input* x ke dalam rentang antara 0 dan 1. Hal ini menjadikan *sigmoid* sangat berguna dalam konteks klasifikasi biner, karena *output* dapat ditafsirkan sebagai probabilitas. Berdasarkan definisi tersebut, grafik fungsi aktivasi *sigmoid* disajikan pada Gambar 12.



Gambar 12. Fungsi Aktivasi *Sigmoid* (Nourbakhsh dan Habibi, 2023).

b. *Softmax*

Softmax menghasilkan *output* dengan rentang nilai antara 0 hingga 1 dengan jumlah peluang yang sama dengan 1. Fungsi aktivasi yang digunakan dalam klasifikasi *multiclass,* bekerja dengan mengubah nilai *output* dari lapisan terakhir menjadi probabilitas sehingga menunjukkan kemungkinan *input* termasuk ke dalam setiap kelas yang tersedia (Obla dkk., 2020). Representasi matematika untuk fungsi aktivasi *softmax* terdapat pada Persamaan (8) (Younisse dkk., 2023).

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_{i=1}^{K} e^{x_i}}, untuk\ j = 1, \dots, K \quad (8)$$

Keterangan:

$\sigma(x_j)$ = Probabilitas prediksi lapisan *output.*

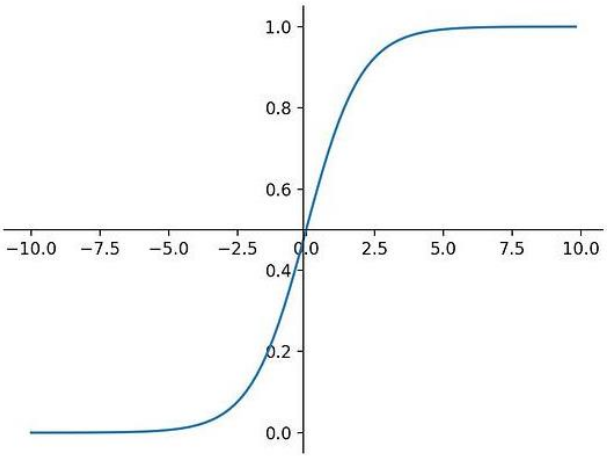
x_i = Sinyal yang dihasilkan lapisan *output.*

e = Euler (2,718).

K = Jumlah kelas lapisan *output.*

$\sum_{i=1}^{n} e^{x_i}$ = Jumlah eksponensial seluruh sinyal *output.*

Berdasarkan Persamaan (8), grafik fungsi aktivasi *softmax* disajikan pada Gambar 13.



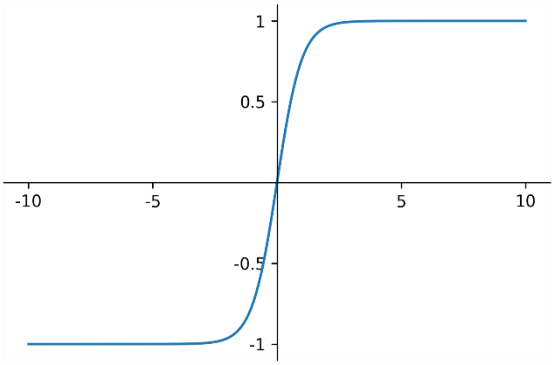
Gambar 13. Fungsi Aktivasi *Softmax* (Nourbakhsh dan Habibi, 2023).

c. *Tangen Hiperbolik* (tanh)

Fungsi aktivasi tanh menghasilkan keluaran terpusat pada nilai nol dengan rentang antara -1 hingga 1 (Shen dkk., 2022). Membantu stabilisasi dengan mengurangi bias karena keluaran positif dan negatif seimbang. Selain itu, fungsi aktivasi tanh dapat mempercepat konvergensi model karena perubahan gradient terukur. Representasi matematika untuk fungsi aktivasi tanh terdapat pada Persamaan (9) (Liu dan Di., 2021).

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{9}$$

Berdasarkan Persamaan (9), grafik fungsi aktivasi tanh dapat dilihat pada Gambar 14.



Gambar 14. Fungsi Aktivasi tanh (Nourbakhsh dan Habibi, 2023).

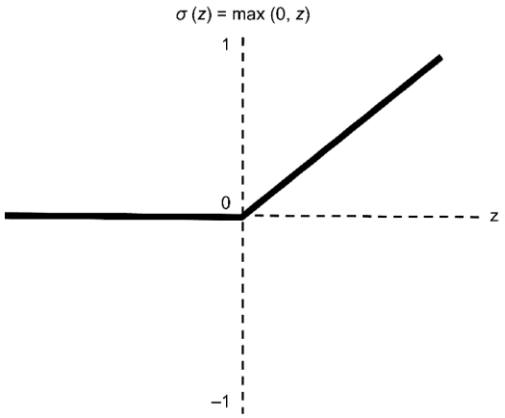
d. *Rectified Linier Unit* (ReLU)

Fungsi aktivasi ReLU merupakan fungsi aktivasi yang memperbaiki nilai *input* kurang dari nol sehingga nilai *output* menjadi nol dan setiap nilai *input* yang

lebih dari nol akan dipertahankan nilainya. ReLU menghasilkan *output* sama dengan *input* saat nilainya positif, tetapi menghasilkan nol saat *input* negatif (Younisse dkk., 2023). Representasi matematika untuk fungsi aktivasi ReLU terdapat pada Persamaan (10).

$$f(x) = max(0, x) \tag{10}$$

Berdasarkan Persamaan (10) berarti bahwa $f(x) = x$ saat $x > 0$, tetapi $f(x) = 0$ saat $x \leq 0$. Grafik fungsi aktivasi ReLU dapat dilihat pada Gambar 15.



Gambar 15. Fungsi Aktivasi ReLU (Nourbakhsh dan Habibi, 2023).

Berdasarkan Gambar 15, dapat diinterpretasikan bahwa fungsi aktivasi ReLU bersifat linier ketika $x > 0$, tetapi pada $x \leq 0$ nilai *output* menjadi nol. Implementasi tersebut menyatakan bahwa fungsi aktivasi ReLU merupakan fungsi *non-linear.* Sifat tersebut yang mampu memperkenalkan *non-linearitas* dengan memodelkan pola kompleks.

e. *Gaussian Error Linier Unit* (GELU)

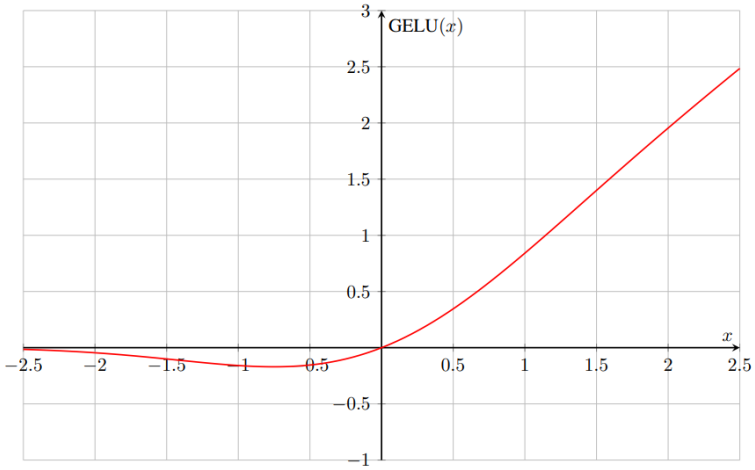
GELU adalah fungsi aktivasi yang menggabungkan keunggulan ReLU dengan hadir lebih stabil. Pada GELU, setiap *input* memiliki kemungkinan untuk diabaikan tergantung pada nilai *input*. Responsibilitas tersebut mengadopsi sifat Gaussian *Cumulative Distribution Function* (CDF) (Roodschild dkk., 2024). Fungsi CDF Gaussian memberikan probabilitas suatu variabel acak mengikuti distribusi normal akan mengimplementasikan nilai kurang dari atau sama dengan x. Secara matematis, fungsi tersebut didefinisikan sesuai Persamaan (11).

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt \tag{11}$$

Penerapan fungsi aktivasi GELU menggabungkan *input* x dengan Gaussian CDF. Respresentasi matematika untuk fungsi aktivasi GELU terdapat pada Persamaan (12) (Lee, 2023).

$$GELU(x) = x \cdot \Phi(\propto x) = 0.5x\left(1 + tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right) \quad (12)$$

Berdasarkan Persamaan (12), grafik fungsi aktivasi GELU dapat dilihat pada Gambar 16.



Gambar 16. Fungsi Aktivasi GELU (Lee, 2023).

Berdasarkan Gambar 16, GELU memiliki bentuk halus tanpa titik tajam yang memungkinkan perubahan mulus antara *output* saat *input* meningkat. Hal tersebut mengakibatkan saat *input* negatif menghasilkan *output* GELU mendekati nol, tetapi tidak benar-benar mencapai nol. Oleh karena itu, GELU menjadi pilihan dalam menangani kompleksitas data dengan hubungan *non-linear.*

2.13 *Deep Learning* (DL)

Menurut Li dkk., (2022), DL merupakan pengembangan dari ML yang memungkinkan model untuk belajar secara otomatis tanpa instruksi eksplisit dari pengguna. Berbeda dari pendekatan tradisional, DL dapat mengenali pola secara mandiri dari data jumlah besar, meniru cara manusia membuat keputusan yang

kompleks. Berdasarkan kemampuan ini, *deep learning* unggul dalam menemukan pola-pola tersembunyi dan hubungan *non-linear* dalam data. *Deep learning* telah menunjukkan performa yang sangat baik dalam berbagai aplikasi, termasuk pengenalan suara, visi komputer, dan pemrosesan bahasa alami. Kemampuan untuk menangani data besar dan kompleks serta fleksibilitas dalam mengekstraksi fitur tanpa perlu rekayasa fitur yang ekstensif, DL telah menjadi teknologi yang sangat berpengaruh dalam pengembangan sistem cerdas.

Kekuatan utama dari DL terletak pada kemampuannya untuk belajar dari data dan menemukan pola yang tidak terlihat oleh metode tradisional, menjadikannya alat berharga dalam berbagai bidang serta lebih unggul dibandingkan metode ML. Menurut Alzubaidi dkk., (2021), mengklasifikasikan DL ke dalam beberapa kategori utama sebagai berikut:

a. *Deep supervised learning*: Teknik ini diterapkan pada data yang berlabel, memungkinkan model untuk belajar dari pengetahuan yang telah ada sebelumnya untuk memprediksi atau menghasilkan keluaran tertentu. Teknik *supervised* ini umumnya lebih sederhana dari metode lain dan mampu mencapai performa tinggi dalam berbagai aplikasi.
b. *Deep semi-supervised learning*: Pendekatan ini digunakan pada data yang sebagian besar tidak berlabel. Tujuan dari teknik ini adalah untuk mengelompokkan data serupa ke dalam satu area tertentu tanpa memerlukan label spesifik. *Semi-supervised* learning mampu mendeteksi pola tersembunyi dalam data tidak berlabel, sehingga sering diterapkan untuk klasifikasi pola.
c. *Deep unsupervised learning*: Teknik ini mengandalkan data yang sepenuhnya tidak berlabel dan mencakup metode seperti pengurangan dimensi dan *clustering*. Kekurangan utama dari metode ini adalah kesulitan dalam menghasilkan informasi yang sangat akurat terkait pengelompokan data dan beban komputasi yang tinggi.
d. *Deep reinforcement learning*: Teknik ini melibatkan pembelajaran dari interaksi langsung dengan lingkungan, berbeda dari *supervised learning* yang bergantung pada data yang diberikan. Tidak adanya fungsi kerugian yang jelas membuat pembelajaran *reinforcement* lebih sulit dibandingkan *supervised*

learning, karena model harus menemukan strategi yang optimal melalui percobaan bertahap.

Sesuai perkembangan teknologi, dalam permasalahan pemrosesan bahasa alami dapat diatasi oleh model DL dengan kinerja yang lebih optimal, yaitu algoritma *Transformer*. *Transformer* menjadi bagian dari DL secara khusus telah banyak digunakan dalam pemrosesan bahasa alami dan pengenalan pola sekuensial, menjadikannya penting dalam banyak aplikasi AI modern.

2.14 *Adam Optimizer*

Optimizer adalah algoritma yang digunakan dalam pembelajaran mesin untuk memperbarui parameter model, seperti bobot dan *learning rate* selama proses pelatihan. Tujuan utama dari *optimizer* adalah untuk meminimalkan fungsi *cost* yang mengukur seberapa baik model. Pada *deep learning* tingkat akurasi optimal dapat diatasi dengan pemilihan model *optimizer* signifikan. Implementasi *optimizer* yang umum dalam *deep learning* ialah *Adam Optimizer* dengan integrasi berbagai kelebihan yang dimiliki. *Adaptive Moment Estimation* atau yang biasa disebut dengan *Adam* memberikan berbagai keunggulan efektivitas dalam integrasinya. Keunggulan implementasi *Adam Optimizer* terutama pada integrasinya dalam mempertimbangkan gradient dan iterasi sebelumnya serta menyesuaikan *learning rate* berdasarkan rata-rata kuadrat gradien (Yi dkk., 2020). Oleh karena itu, integrasi implementasi *Adam Optimizer* menyebabkan nilai *learning rate* dapat berubah secara dinamis dalam proses pelatihan serta menghasilkan kinerja model optimal (Soydaner, 2020).

2.15 *Hyperparameter Tuning*

Pada beberapa algoritma pembelajaran memiliki parameter yang dapat disesuaikan karena pengaturan parameter yang berbeda menghasilkan model dengan kinerja yang beragam. Evaluasi dan pemilihan model tidak hanya berkaitan dengan memilih algoritma pembelajaran, tetapi juga dengan mengoptimalkan parameter-

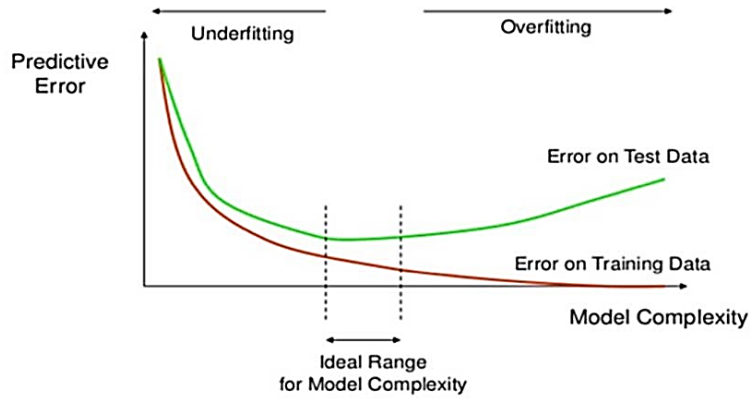
parameter tersebut. Proses mencari nilai optimal untuk parameter-parameter ini dikenal sebagai *hyperparameter tuning* (Isabona dkk., 2022). Pada pembelajaran mesin terdapat parameter utama yang dapat disesuaikan untuk mengoptimalkan kinerja pada tahap *hyperparamater tuning*, yaitu *batch size*, *learning rate*, jumlah *epoch*, dan *dropout* (Khairani dkk., 2024).

Nilai *learning rate* biasanya berkisar antara 0 dan 1, bervariasi dari kasus ke kasus (Priyanka dkk., 2024). *Learning rate* yang terlalu rendah dapat memperlambat proses konvergensi karena model membutuhkan waktu lebih lama untuk mencapai nilai optimal. Akan tetapi, proses tersebut mengintegrasikan pendekatan optimum global dengan akurasi validasi yang lebih baik. Sehingga model mampu menghindari *overfitting* serta mempertahankan generalisasi. Berlawanan dengan *learning rate* terlalu tinggi menghasilkan pelatihan berlangsung cepat dan model dapat mencapai akurasi data latih yang tinggi. Namun, hal tersebut menyebabkan ketidakstabilan dalam pelatihan karena membuat model sulit mencapai titik konvergen karena sering melewati optimum lokal (Lewkowycz dkk., 2020). Oleh karena itu, dalam mencapai hasil terbaik penting untuk hati-hati memilih nilai *learning rate*.

Kemudian jumlah *epoch* yang perlu diperhatikan dengan baik dalam *hyperparamater tuning*. Jumlah *epoch* menentukan seluruh dataset pelatihan dilakukan untuk memperbarui bobot model (Shin dkk., 2020). *Batch size* adalah jumlah sampel yang diintegrasikan pelatihan model dalam setiap *epoch* (Kandel dan Castelli, 2020). Nilai *batch size* yang besar dapat meningkatkan performa model dengan mencapai titik minimum optimal, lebih baik dari nilai *batch size* kecil. Namun, dengan nilai *batch size* yang besar memerlukan proses komputasi dan kapasitas memory lebih besar. Selanjutnya, *dropout* merupakan teknik efektif yang mampu mencegah *overfitting* dengan membantu model berukuran besar melakukan pelatihan pada data pelatihan yang tidak cukup besar (Ramadhan dkk., 2024).

Overfitting adalah kondisi ketika model belajar terlalu baik dari data pelatihan hingga menangkap *noise* atau pola yang tidak relevan, sehingga performa pada data pelatihan sangat tinggi dan buruk pada data baru. Sedangkan, *underfitting* adalah

kondisi ketika model tidak mampu mengenali pola dalam data pelatihan sehingga menghasilkan performa buruk baik pada data pelatihan maupun data uji (Al-Behadili dkk., 2018). *Overfitting* dan *underfitting* diilustrasikan pada Gambar 17.



Gambar 17. Ilustrasi *Overfitting* dan *Underfitting* (Al-Behadili dkk., 2018).

Berdasarkan Gambar 17, kondisi tersebut merupakan tantangan umum dalam pelatihan model dan menjadi alasan pentingnya penyesuaian *hyperparameter* secara tepat. Pencarian nilai *hyperparameter* optimal, dapan menggunakan teknik memeriksa beberapa kombinasi nilai *hyperparameter* pengujian. Menurut Elgeldawi dkk., (2021), *hyperparamater tuning* umumnya menggunakan algoritma *grid search, random search,* dan *Bayesian search.*

a. *Grid Search*

   Grid Search adalah pendekatan yang mengevaluasi setiap kemungkinan kombinasi *hyperparameter* dalam ruang pencarian yang tetap. Teknik ini melatih model menggunakan setiap konfigurasi *hyperparameter*, kemudian memilih konfigurasi yang menghasilkan performa terbaik.

b. *Random Search*

   Random Search mengambil pendekatan yang lebih acak dengan memilih beberapa kombinasi *hyperparameter* secara acak dalam ruang pencarian. Sehingga cara kerja yang dihasilkan acak mengevaluasi titik tertentu dalam ruang pencarian untuk menemukan konfigurasi yang optimal.

c. *Bayesian Search* (BS)

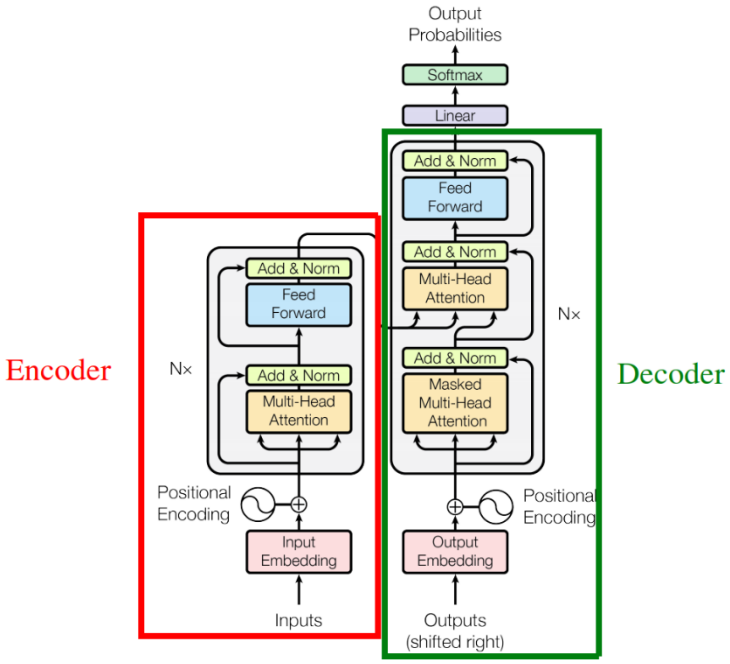
   Bayesian Search mengintegrasikan pendekatan berbeda dengan memanfaatkan hasil dari setiap iterasi sebelumnya untuk memandu pencarian *hyperparameter*

berikutnya. *Bayesian Search* menafsirkan proses tuning *hyperparameter* sebagai masalah optimisasi fungsi *black-box*.

2.16 *Transformer*

Transformer merupakan arsitektur *deep learning* yang dikembangkan oleh para peneliti di Google pada tahun 2017 (Vaswani dkk., 2017). *Transformer* dalam pengembangannya mengimplementasikan *attention mechanism* tanpa menggunakan dasar *recurrent* dan konvolusi. Integrasi tersebut mengembangkan kelemahan yang dimiliki *recurrent* dalam paralelisasi yang sebelumnya dalam *recurrent* hanya mengandalkan *output* dari satu langkah untuk memproses langkah berikutnya. Pada integrasi *Transformer* telah mengadopsi sistem paralel, sehingga model dapat memperhatikan seluruh urutan sekaligus tanpa harus menunggu hasil dari langkah sebelumnya yang memungkinkan pelatihan model jauh lebih cepat.

Kinerja tersebut dihasilkan oleh *Transformer* dengan inovasi arsitekturnya, yaitu *self-attention* (Hao dkk., 2021). *Self-attention* merupakan mekanisme yang mampu menghubungkan posisi kata yang berbeda dari kalimat menghasilkan representasi konteks kalimat tersebut dengan memberikan bobot pada elemen-elemen lain dalam urutan yang sama sehingga menangkap hubungan kontekstual antar elemen secara efisien. *Transformer* dapat diimplementasikan pada berbagai tugas NLP, yaitu pemahaman bacaan, klasifikasi teks, hingga pengenalan entitas (NER). Arsitektur *Transformer* dapat dilihat pada Gambar 18.



Gambar 18. Arsitektur Model *Transformer* (Vaswani dkk., 2017).

Berdasarkan Gambar 18, *Transformer* memiliki arsitektur dengan struktur *encoder* dan *decoder*. *Encoder* sebagai penerima urutan *input* $X = (x_1, \dots, x_N)$ yang terdiri dari elemen-elemen *token* dalam urutan *input*. *Encoder* memproses urutan menghasilkan *latent representation* $Z = (z_1, \dots, z_N)$ yang menggambarkan konteks atau informasi dalam urutan *input*. Kemudian *decoder* mengambil representasi tersebut untuk menghasilkan urutan *output* $Y_M = (y_1, \dots, y_M)$, dengan M merupakan panjang urutan *output*.

Decoder menghasilkan urutan *output* satu elemen pada satu waktu dengan *autoregressive* sehingga setiap kata Y_M dihasilkan menggunakan *latent representation* Z dan juga urutan kata-kata sebelumnya $Y_{M-1} = (y_1, \dots, y_{M-1})$ (Gillioz dkk., 2020). Pada *encoder* dan *decoder* masing-masing terdiri dari 6 lapisan identik. Setiap lapisan encoder memiliki dua *sub*-lapisan utama, yaitu *multi-head self-attention* yang memungkinkan model untuk memperhatikan seluruh urutan *input* secara paralel dan *feed-forward* yang menerapkan transformasi *non-linear*. Kemudian dalam *decoder* memiliki tiga *sub*-lapisan, yaitu *masked multi-head self-attention* untuk mencegah kebocoran informasi selama proses *autoregressive*, *multi-head attention* yang menghubungkan *decoder* dengan *output* dari *encoder*, dan *feed-forward* (Vaswani dkk., 2017). Selain itu, kedua bagian ini menggunakan

layer normalization dan *residual connections* untuk stabilitas dan aliran informasi yang lebih baik. Oleh karena itu, terdapat total 12 lapisan dalam *Transformer* dengan penjelasan strukturnya secara umum sebagai berikut.

a. *Embedding*

Proses *embedding* pada *Transformer* mengonversi kata-kata dalam urutan menjadi representasi vektor. *Embedding* memungkinkan model mengenali makna semantik dari kata dalam konteks tertentu pada *input* dan *output* (Liu dkk., 2023). Proses *embedding* untuk token menghasilkan vektor berdimensi yang direpresentasikan pada Persamaan (13).

$$Embedding(x_i) \in \mathbb{R}^{d_{model}} \tag{13}$$

Pada d_{model} dimensi dari vektor *embedding* yang umum digunakan adalah 512. Pada arsitektur *Transformer, embedding* dijumlahkan dengan *Positional Encoding* untuk menyertakan informasi urutan.

b. *Positional Encoding*

Transformer tidak memiliki mekanisme urutan secara intrinsik karena tidak mengintegrasikan struktur *recurrent.* Pada pengembangannya implementasi tersebut diintegrasikan *Transformer* dengan menyertakan informasi tentang urutan dalam posisi relatif menggunakan *positional encoding*. Integrasi tersebut diimplementasikan pada *encoder* dan *decoder.* Pada efisiensinya, *positional encoding* ditambahkan ke setiap vektor *embedding* dengan mengharuskan dimensi keduanya sama (Chen dkk., 2021). *Positional encoding* menggunakan fungsi *sinusoidal* dengan komponen *sinus* dan *kosinus* yang direpresentasikan pada Persamaan (14) dan (15).

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{14}$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{15}$$

Berdasarkan persamaan (14) dan (15), menggambarkan representasi *positional encoding* dengan pos merujuk posisi token dalam urutan dan i adalah indeks dimensi dalam *embedding*. Setiap dimensi dalam *positional encoding* dipetakan ke fungsi *sinus* atau *cosinus* tertentu. Fungsi *sinusoidal* ini memiliki panjang gelombang yang mengikuti progresi geometris, dengan nilai yang berkisar

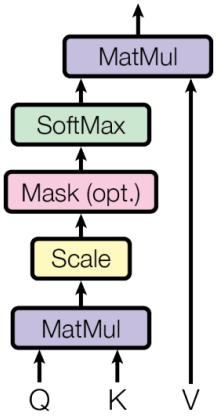
antara 2π hingga $10000 \cdot 2\pi$. Integrasi tersebut memudahkan model mempelajari *mechanism attention* dengan cara efisien dan jelas memahami hubungan antar posisi dalam urutan (Vaswani dkk., 2017).

c. *Attention*

Peran sentral dari hubungan antara *query* dan *key-value* dalam fungsi *attention* telah dieksplorasi lebih lanjut oleh beberapa studi. Implementasi *attention* melalui ntegrasi perhitungan bobot berbasis kesamaan antara *query* dan *key* meningkatkan kemampuan model dalam menangkap hubungan antar elemen dalam data sekuensial sesuai relevansi. Oleh karena itu, integrasi fungsi *attention* dapat meningkatkan efisiensi dan kemampuan model dalam menangkap hubungan kompleks data sekuensial.

d. *Scaled Dot-Product Attention*

Scaled dot-product attention adalah mekanisme dalam model *Transformer* yang bekerja dengan tiga himpunan vektor utama, yaitu *query* (Q), *key* (K), dan *value* (V). Dimensi vektor tersebut secara berurutan adalah $d_q, d_k,$ dan d_v. Representasi vektor tersebut dalam bentuk matriks $Q = [q_1, q_2, \dots, q_n]^T$, $K = [k_1, k_2, \dots, k_n]^T$, dan $V = [v_1, v_2, \dots, v_n]^T$. Proses *scaled dot-product attention* dengan menghitung perkalian skalar antara vektor *query* dan *key* untuk setiap token dalam urutan. Kemudian hasil perkalian tersebut dibagi dengan $\sqrt{d_k}$, untuk menghasilkan vektor *value* dari peroleh bobot diterapkan fungsi *softmax* dalam implementasinya. Proses tersebut merupakan integrasi kemampuan unggulan *Transformer,* yaitu *self-attention*. Representasi proses tersebut dapat dilihat pada Gambar 19.



Gambar 19. *Scaled Dot-Product Attention* (Vaswani dkk., 2017).

Berdasarkan Gambar 19, diformulasikan perhitungan dalam proses *Scaled dot-product attention* pada Persamaan (16).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (16)$$

Keterangan:

Q = *Query* representasi *input* untuk mencari hubungan antar token.

K = *Key* representasi *input* untuk mencocokkan *query.*

V = *Value* representasi *input* yang akan dihitung atensinya.

d_k = Dimensi dari *key.*

e. *Multi-Head Attention*

Multi-head attention diimplementasikan dalam *Transformer* dengan tujuan mengintegrasikan beberapa *mechanism attention* dalam satu waktu, yang membantu model menangkap berbagai aspek informasi lebih baik. Proses implementasi tersebut menghasilkan h sebanyak *attention* dalam setiap token yang berbeda sehingga menangkap hasil keberagaman (Chen dkk., 2021). Formulasi proses *multi-head attention* direpresentasikan pada Persamaan (17) dan (18).

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (17)$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (18)$$

Pada W_i^Q, W_i^K, W_i^V yang merupakan matriks proyeksi untuk setiap *head*, dan W^O adalah matriks proyeksi akhir (Chen dkk., 2021). Teknik ini memungkinkan *mechanism attention* dibagi menjadi beberapa *head* untuk beroperasi pada ruang representasi yang berbeda.

f. *Feed-Forward Network*

Feed-Forward Network (FFN) pada *Transformer* yang terdapat pada tiap lapisan *encoder* dan *decoder* adalah lapisan yang melakukan transformasi *linear* pada setiap token secara terpisah. FFN pada *Transformer* merupakan jaringan *fully connected* terdiri dari dua lapisan *linear* dengan aktivasi ReLU yang menghubungkan keduanya (Vaswani dkk., 2017). Representasi FFN dapat dilihat pada Persamaan (19).

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (19)$$

Keterangan:

W = matriks bobot

b = bias

g. *Fully Connected Layer*

Fully connected layer atau *linear layer* sebagai tahap akhir sebelum menghasilkan prediksi. Lapisan ini bertanggung jawab untuk memproyeksikan vektor keluaran dari *Transformer* ke dimensi *output* yang sesuai dengan tugas yang dihadapi. Representasi *linear layer* dapat dilihat pada Persamaan (20).

$$z = Wh + b \tag{20}$$

Keterangan:

h = vektor fitur model

Lapisan ini berfungsi sebagai pemetaan akhir dari fitur yang telah diekstraksi sebelumnya ke dimensi jumlah kelas atau label target. Jika model digunakan untuk klasifikasi, nilai *logits* yang diperoleh dari lapisan ini akan diteruskan ke fungsi *softmax* untuk menghasilkan probabilitas prediksi.

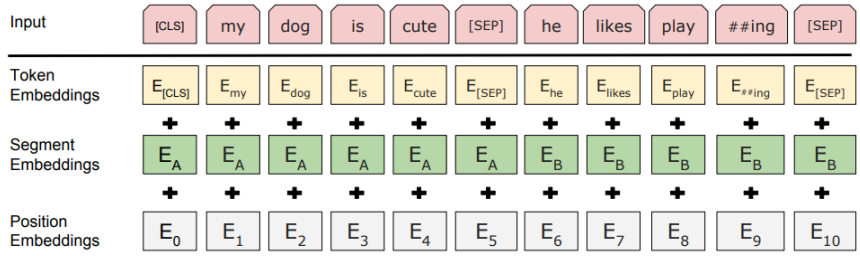
Penggunaan *Transformers* mendorong pengembangan *pre-trained systems*, seperti *Large Language Model* Meta AI (LLaMA) dan *Bidirectional Encoder Representations from Transformers* (BERT). Pada integrasinya, LLaMA mengadopsi pendekatan *decoder* dari *Transformer* yang berfokus pada *Causal Language Modeling* (CLM) secara *autoregressive*. Sementara itu, model BERT memanfaatkan pendekatan *encoder* dari *Transformer* dengan fokus integrasi pada *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP).

2.17 *Bidirectional Encoder Representations from Transformers*

Model BERT adalah sebuah representasi *encoder* dari model *Transformer*, sebuah arsitektur NLP yang memanfaatkan mekanisme perhatian untuk menggantikan *recurrent* sehingga mampu menangkap hubungan kontekstual antara kata-kata yang berjauhan. Pendekatan model BERT adalah salah satu *embedding* teks canggih yang dikeluarkan oleh Google (Ravichandiran, 2021). Model BERT merupakan representasi berbasis *fine-tuning* pertama yang mencapai kinerja terbaik pada

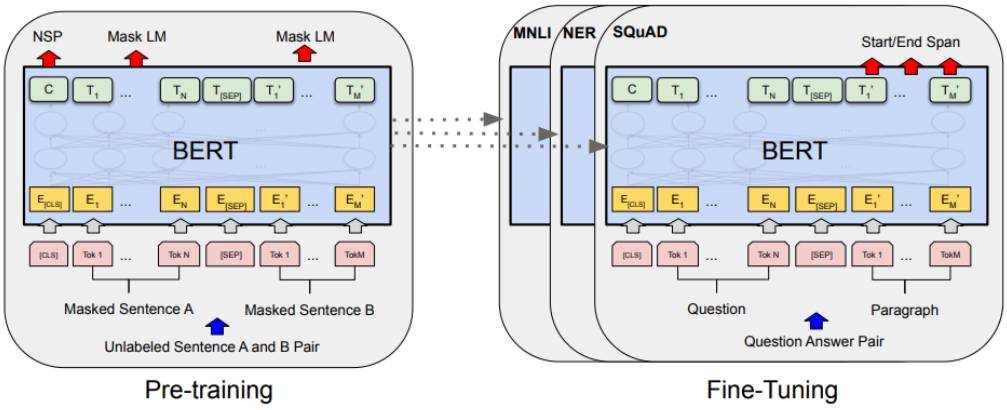
berbagai tugas besar *sentence-level* dan *token-level* serta melampaui banyak arsitektur yang dirancang khusus untuk tugas tertentu (Devlin dkk., 2019). Model BERT bermanfaat bagi NLP seperti menjawab pertanyaan, pembuatan teks, dan klasifikasi teks. Model BERT menghasilkan representasi untuk setiap kata dalam kalimat sebagai *output* (Chandradev dkk., 2023).

Pada implementasi model BERT mengintegrasikan 2 token, yaitu [CLS] dan [SEP]. Proses tersebut diawali dengan token [CLS] sebagai token klasifikasi serta diakhiri [SEP] untuk token yang memisahkan antar konteks kalimat. Proses *input embedding* dalam *encoder* BERT mengintegrasikan penjumlahan token *embedding*, *segment embedding,* dan *positional embedding*. Representasi struktur tersebut dapat dilihat pada Gambar 20.



Gambar 20. Representasi *Input* Model BERT (Devlin dkk., 2019).

Berdasarkan Gambar 20, representasi *input* model BERT dilanjutkan untuk implementasi dua arsitektur mulai dari *pre-training* hingga *fine-tuning*. Pada keduanya melibatkan arsitektur sama dengan perbedaan pada masing-masing *output*. Representasi integrasi tersebut dapat dilihat pada Gambar 21.



Gambar 21. Arsitektur Model BERT (Devlin dkk., 2019).

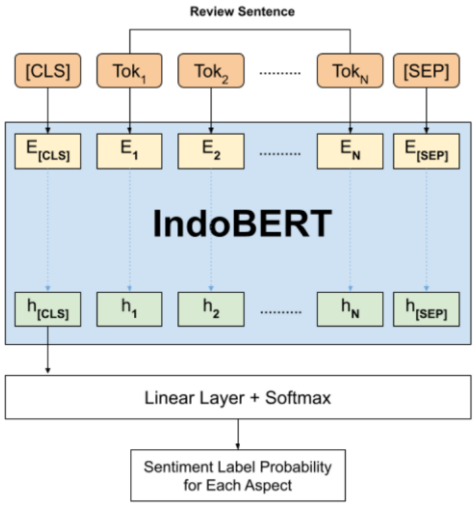
Berdasarkan Gambar 21, arsitektur model BERT tidak hanya mengadopsi kemampuan *bidirectional*, tetapi juga mengintegrasikan dua teknik pelatihan utama. Integrasi tersebut merupakan bagian dari proses *Masked Language Modeling* (MLM) serta *Next Sentence Prediction* (NSP). Pada MLM sebagian kata dalam kalimat disembunyikan dan model dilatih untuk memprediksi kata yang hilang berdasarkan konteks sekitarnya. Implementasi tersebut diintegrasikan untuk setiap *input,* 15% token dalam kalimat dipilih secara acak akan disembunyikan (Ouellette dan Rudzicz, 2020).

Proses selanjutnya, model BERT akan memprediksi kata-kata yang disembunyikan tersebut secara simultan. Sementara itu, NSP tugas tambahan yang signifikan digunakan membantu model memahami hubungan antara dua kalimat. NSP model dilatih untuk memprediksi dua kalimat yang diberikan saling berurutan atau tidak dalam sebuah teks. Tugas ini membantu model BERT dalam memahami hubungan semantik antar kalimat. Token [CLS] diletakkan di awal *input* untuk mewakili keseluruhan kalimat. Pada proses tersebut selama pelatihan, 50% dari pasangan kalimat adalah pasangan sebenarnya, tetapi 50% lainnya adalah kalimat yang diambil secara acak dari korpus (Devlin dkk., 2019). Seluruh integrasi tersebut membuat arsitektur model BERT lebih optimal.

2.18 *Indonesia Bidirectional Encoder Representations from Transformers*

Model IndoBERT merupakan *pre-trained* model berbasis *Transformer* menggunakan arsitektur model BERT yang dilatih khusus untuk bahasa Indonesia. Model ini dilatih sebagai model bahasa *masked* dengan menggunakan *framework Huggingface* (Koto dkk., 2020). Proses pelatihan model IndoBERT menggunakan *dataset* Indo4B yang memiliki lebih dari 23 GB data teks bahasa 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 dkk., 2020). Akar model IndoBERT adalah *Transformer* yang memiliki arsitektur identik dengan model BERT sehingga memungkinkan integrasi proses

fine-tuning atau melatih data baru dalam menciptakan model sesuai *dataset* spesifik. Integrasi model IndoBERT juga mengadopsi tugas utama dalam pelatihan model BERT, yaitu MLM dan NSP. Representasi arsitektur model IndoBERT dapat dilihat pada Gambar 22.



Gambar 22. Arsitektur Model IndoBERT (Yulianti dan Nissa, 2024).

Berdasarkan Gambar 22, struktur model IndoBERT mengadopsi model BERT yang terdiri dari tumpukan *encoder* arsitektur *Transformer*. Model IndoBERT memiliki dua varian arsitektur, yaitu IndoBERT *base* dan IndoBERT *large*. Keduanya berbeda dalam jumlah lapisan *Transformer*, *attention head*, dan jumlah parameter. Model IndoBERT *base* memiliki 12 lapisan *Transformer*, 12 *attention head*, dan 110 juta parameter. Sedangkan, model IndoBERT *large* memiliki 24 lapisan *Transformer*, 16 *attention head*, dan 340 juta parameter. Pada implementasi keduanya, model IndoBERT *large* umumnya memberikan presisi yang lebih tinggi karena dapat menangkap dengan lebih baik semantik teks sebagai hasil dari penggunaan jumlah parameter optimal. Namun, pelatihan data dengan model IndoBERT *large* biasanya membutuhkan waktu lebih lama daripada model IndoBERT *base* (Geni dkk., 2023). Model IndoBERT yang telah dilatih dengan data bahasa Indonesia yang besar dan spesifik mampu mengatasi tugas NLP terutama klasifikasi teks dengan lebih baik dibandingkan model BERT karena *pre-trained* yang optimal yang memfasilitasi pemahaman konteks dan struktur bahasa Indonesia secara lebih mendalam (Nabiilah dkk., 2023).

2.19 *Distilled Bidirectional Encoder Representations from Transformers*

Model DistilBERT merupakan modifikasi hasil kompresi dari model BERT. Model DistilBERT memanfaatkan proses *knowledge distillation* untuk mengurangi ukuran model serta mempercepat proses inferensi tanpa mengurangi akurasi. *Knowledge distillation* dalam implementasinya mengadopsi alur proses model ukuran yang lebih kecil sebagai *student model* dilatih untuk mereplikasi kehandalan model besar yang disebut *teacher model*. Pada *supervised learning,* model klasifikasi umumnya dilatih memprediksi kelas dari suata data dengan memaksimalkan probabilitas yang diperkirakan terhadap label yang tepat (Basbeth dan Fudholi, 2024).

Tujuan utama pelatihan meminimalkan *cross-entropy* antara distribusi yang diprediksi oleh model dan distribusi *one-hot* dari label yang benar. Sebuah model yang performanya baik akan memberikan probabilitas tinggi untuk kelas yang benar dan probabilitas kecil untuk kelas lainnya (Sanh dkk., 2020). Berdasarkan hal tersebut selama efisiensi implementasinya, model DistilBERT meniru distribusi probabilitas dari model BERT menilai setiap kelas termasuk probabilitas kecil untuk kelas yang salah karena pembelajaran tersebut membantu model DistilBERT dalam proses distilasi. Adopsi pembelajaran tersebut diformulasikan dengan representasi dari *distillation loss* pada Persamaan (21).

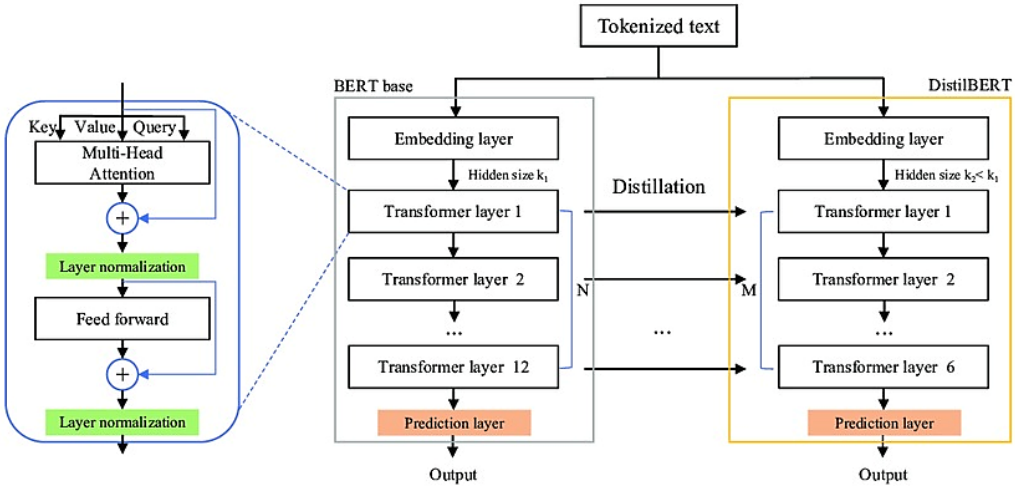
$$Lce = \sum_{i} t_i * \log(si) \tag{21}$$

Pada t_i yang merupakan probabilitas target dari model BERT dan si merupakan probabilitas prediksi dari model DistilBERT. Kemudian proses distilasi juga menggunakan teknik *softmax-temperature* dengan representasi pada Persamaan (22).

$$p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j exp\,(z_j/T)} \tag{22}$$

Nilai z_i merupakan skor yang diberikan oleh model untuk kelas ke-i. Integrasi *softmax-temperature* dalam model DistilBERT membuat model kecil lebih komprehensif dengan tidak hanya meniru kelas probabilitas tinggi, tetapi juga memahami hubungan antar kelas (Adel dkk., 2022). Dengan demikian, pengembangan model BERT pada model DistilBERT fokus terhadap integrasi

distillation loss dengan *supervised training loss* serta menghapus NSP mengurangi kompleksitas pelatihan. Representasi arsitektur model DistilBERT dapat dilihat pada Gambar 23.



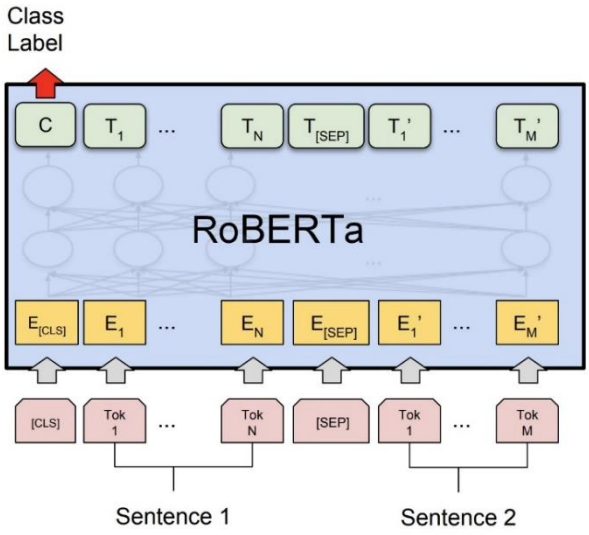
Gambar 23. Arsitektur Model DistilBERT (Adel dkk., 2022).

Berdasarkan Gambar 23, dapat dilihat bahwa model DistilBERT menggunakan *encoder* pada *Transformer.* Jumlah *layer* BERT *base* pada *encoder* yang berjumlah 12 *layer Transformer* diadopsi model DistilBERT hanya memiliki 6 *layer Transformer.* Ukuran *hidden layer* pada model DistilBERT lebih kecil atau sama dengan model BERT. Salah satu pengembangan model DistilBERT adalah model DistilBERT *Multilingual* yang dilatih menggunakan Wikipedia dengan 104 bahasa berbeda, termasuk bahasa Indonesia. Model tersebut memiliki 6 *layer,* 768 dimensi, 12 *attention heads,* serta 134 juta parameter (Adel dkk., 2022). Spesifikasi arsitektur *pre-trained* tersebut menjadikan implemnetasi model DistilBERT lebih kecil dan efisien cocok diterapkan dalam permasalahan dengan sumber daya komputasi yang terbatas.

2.20 *Robustly Optimized Bidirectional Encoder Representations from Transformers Pretraining Approach*

Model RoBERTa adalah versi yang dioptimalkan dari model BERT, dengan beberapa integrasi pada *pre-trained.* Model RoBERTa mengadopsi arsitektur *encoder* pada *Transformer* sama seperti model BERT. Model RoBERTa merupakan pengembangan signifikan dari model BERT dengan menghilangkan

tugas NSP, integrasi *dynamic masking, batch* yang lebih besar, serta *Byte-Level Byte Pair Encoding* (BPE) yang lebih besar meningkatkan kemampuan model memproses kata yang belum pernah terlihat sebelumnya (Liu dkk., 2019). Berdasarkan pengembangan tersebut, arsitektur model RoBERTa direpresentasikan seperti pada Gambar 24.



Gambar 24. Arsitektur Model RoBERTa (Liu dkk., 2019).

Berdasarkan Gambar 24, arsitektur model RoBERTa *base* memiliki 12 *layer* dan *12 attention heads*. Sedangkan model RoBERTa *large* memiliki 24 *layer* dan 16 *attention heads*. Salah satu pengembangan model RoBERTa adalah *Cross-lingual Language* Model - *Robustly Optimized Bidirectional Encoder Representations from Transformers Pretraining Approach* (XLM-RoBERTa) yang dilatih menggunakan Wikipedia dengan 100 bahasa berbeda, termasuk bahasa Indonesia. Terdapat dua pengembangan dari model XLM-RoBERTa, yaitu XLM-RoBERTa *base* dan XLM-RoBERTa *large.* Pada model XLM-RoBERTa *base* mengimplementasikan arsitektur 12 *layer,* 768 dimensi, 12 *attention heads,* serta 270 juta parameter. Sedangkan model XLM-RoBERTa *large* dengan 24 *layer,* 1024 dimensi, 16 *attention heads,* serta 550 juta parameter (Conneau dkk., 2020).

2.21 Evaluasi Model

Evaluasi kinerja model merupakan langkah penting dalam proses klasifikasi dengan pembelajaran mesin untuk menilai seberapa efektif model yang telah dilatih dalam

mengenali pola dan memprediksi hasil pada data baru yang tidak pernah terlihat sebelumnya. Proses evaluasi ini membantu dalam memastikan bahwa model tidak hanya bekerja baik pada data pelatihan, tetapi juga memiliki kemampuan untuk menggeneralisasi data di luar sampel yang dilatih dengan mengurangi risiko *overfitting*. Evaluasi dapat mengukur kinerja model yang dihasilkan dari proses *fine-tuning* (Khairani dkk., 2024).

Pada konteks klasifikasi, evaluasi yang dihasilkan oleh model merupakan probabilitas dari suatu kelas tertentu berdasarkan fitur yang diamati. Pendekatan probabilistik ini dapat dijelaskan menggunakan Teorema Bayes, yang menyatakan bahwa probabilitas suatu kejadian berdasarkan hasil klasifikasi. Jika terdapat suatu himpunan kelas $C_0, C_1, \dots, C_N$ yang saling tidak beririsan (partisi dari ruang sampel S) dengan probabilitas prior pada Persamaan (23):

$$P(C_0), P(C_1), \dots, P(C_N) \tag{23}$$

Pada $P(C_i) \neq 0$ untuk setiap $i = 0, 1, \dots, N$. Jika model menghasilkan suatu hasil klasifikasi B, maka probabilitas posterior dari kelas C_i, setelah mempertimbangkan hasil klasifikasi B dapat dinyatakan oleh Persamaan (24) (Magni dkk., 2002):

$$P(C_i|B) = \frac{P(C_i)P(B|C_i)}{\sum_{j=1}^{k} P(C_j)P(B|C_j)} \tag{24}$$

Keterangan:

$P(C_i|B)$ = probabilitas posterior dari kelas $C_i$setelah memperhitungkan kejadian B.

$P(C_i)$ = probabilitas prior sebelum hasil klasifikasi B diperoleh.

$P(B|C_i)$ = likelihood peluang hasil klasifikasi B jika C_i benar.

Salah satu metode umum untuk mengevaluasi performa model klasifikasi adalah menggunakan *confusion matrix*, yaitu matriks yang menyajikan informasi hasil prediksi model dalam empat nilai utama sebagai berikut:

a. *True Positive* (TP): Jumlah sampel yang diprediksi positif oleh model dan sebenarnya memang positif.
b. *False Positive* (FP): Jumlah sampel yang diprediksi positif oleh model, tetapi sebenarnya negatif.
c. *False Negative* (FN): Jumlah sampel yang diprediksi negatif oleh model, tetapi sebenarnya positif.

d. *True Negative* (TN): Jumlah sampel yang diprediksi negatif oleh model dan sebenarnya memang negatif.

Confusion matrix dasarnya merupakan tabel dimensi 2×2 yang mewakili kelas prediksi (*predicted class)* dan kelas sebenarnya (*actual class*) yang berguna dalam model klasifikasi biner. *Confusion matrix* untuk klasifikasi biner tidak sepenuhnya berlaku pada klasifikasi *multiclass*. *Confusion matrix* pada klasifikasi *multiclass* memiliki dimensi $N \times N$, untuk N merupakan jumlah label kelas yang berbeda dengan $(C_0, C_1, \dots, C_N)$. Oleh karena itu, pada *confusion matrix* klasifikasi *multiclass* menggabungkan matriks yang digunakan dalam klasifikasi biner. Pada klasifikasi biner dan *multiclass* representasi ditunjukkan pada Gambar 25.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

		Predicted Class			
		C_1	**C_2**	**…**	**C_N**
Actual Class	**C_1**	$C_{1,1}$	FP	…	$C_{1,N}$
	C_2	FN	TP	…	FN
	…	…	…	…	…
	C_N	$C_{N,1}$	FP	…	$C_{N,N}$

Gambar 25. *Confusion Matrix* Klasifikasi Biner dan *Multiclass* (Markoulidakis dkk., 2021).

Berdasarkan Gambar 25, beberapa metrik yang optimal digunakan dalam tugas klasifikasi *multiclass*, seperti *accuracy*, *precision*, *recall*, *F1-score,* dan ROC AUC. Pada klasifikasi kejadian, kejadian A_i dapat mewakili kelas yang ingin diklasifikasi, sedangkan B merupakan hasil klasfikasi model. Oleh karena itu, metrik tersebut seluruhnya memberikan wawasan mendalam tentang aspek-aspek berbeda dari performa model yang dapat direpresentasikan dengan pendekatan probabilistik menggunakan Teorema Bayes. Berdasarkan implementasi *confusion matrix*, dapat diintegrasi formula beberapa evaluasi utama meliputi:

1. *Accuracy*

   Accuracy dalam tugas klasifikasi mengukur seberapa optimal model memprediksi data ke dalam kategori yang tepat, baik untuk kelas positif maupun negatif. Representasi matematika untuk *accuracy* dapat dilihat pada Persamaan (25) (Markoulidakis dkk., 2021).

$$Accuracy = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}} \quad (25)$$

Pendekatan konsep Teorema Bayes untuk menghitung probabilitas klasifikasi benar dengan Persamaan (26) hingga (28).

$$Accuracy_{Bayesian} = \sum_{i=1}^{N} P\left(\hat{Y} = C_i \middle| Y = C_i\right) P(Y = C_i) \quad (26)$$

$$P\left(\hat{Y} = C_i \middle| Y = C_i\right) = \frac{TP(C_i)}{\sum_{j=1}^{N} C_{i,j}} \quad (27)$$

$$P(Y = C_i) = \frac{\sum_{j=1}^{N} C_{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}} \quad (28)$$

2. *Precision*

Precision atau *Positive Predictive Value* (PPV) mengukur tingkat ketepatan prediksi positif model, yaitu seberapa banyak prediksi positif yang benar-benar positif dan relevan. *Precision* penting saat prediksi positif memiliki konsekuensi yang kompleks. Representasi matematika untuk *precision* dapat dilihat pada Persamaan (29) hingga (31) (Markoulidakis dkk., 2021).

$$Precision = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \quad (29)$$

$$Precision\ (macro\ average) = \frac{1}{N}\sum_{i=1}^{N} Precision\ (C_i) \quad (30)$$

$$Precision\ (micro\ average) = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N}[TP(C_i) + FP(C_i)]} \quad (31)$$

Pendekatan konsep Teorema Bayes untuk intepretasi *precision* dapat di representasikan dengan Persamaan (32) hingga (36).

$$Precision_{Bayesian} = P(C_i|B) = \frac{P(B|C_i)P(C_i)}{P(B|C_i)P(C_i) + P(B|\bar{C}_i)P(\bar{C}_i)} \quad (32)$$

$$P(C_i) = \frac{TP(C_i) + FN(C_i)}{TP(C_i) + FP(C_i) + FN(C_i) + TN(C_i)} \quad (33)$$

$$P(B|C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (34)$$

$$P(\bar{C}_i) = \frac{FP(C_i) + TN(C_i)}{TP(C_i) + FP(C_i) + FN(C_i) + TN(C_i)} \quad (35)$$

$$P(B|\bar{C}_i) = \frac{FP(C_i)}{FP(C_i) + TN(C_i)} \quad (36)$$

3. *Recall*

Recall atau *True Positive Rate* (TPR) mengukur seberapa baik model dalam mendeteksi semua data positif yang ada. *Recall* meminimalkan FP memastikan bahwa data positif terdeteksi oleh model (Hicks dkk., 2022). Representasi matematika untuk *recall* dapat dilihat pada Persamaan (37) hingga (39) (Markoulidakis dkk., 2021).

$$Recall = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (37)$$

$$Recall\ (macro\ average) = \frac{1}{N}\sum_{i=1}^{N} Recall\ (C_i) \quad (38)$$

$$Recall\ (micro\ average) = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N}[TP(C_i) + FN(C_i)]} \quad (39)$$

Pendekatan konsep Teorema Bayes untuk intepretasi *recall* dapat di representasikan dengan Persamaan (40) hingga (42) (Jamart, 1992).

$$Recall_{Bayesian} = P(B|C_i) = \frac{P(C_i \cap B)}{P(C_i)} \quad (40)$$

$$P(C_i \cap B) = \frac{TP(C_i)}{TP(C_i) + FP(C_i) + FN(C_i) + TN(C_i)} \quad (41)$$

$$P(C_i) = \frac{TP(C_i) + FN(C_i)}{TP(C_i) + FP(C_i) + FN(C_i) + TN(C_i)} \quad (42)$$

4. *F1-score*

F1-score adalah rata-rata harmonic dari *precision* dan *recall*. Integrasi dalam mencapai keseimbangan antara *precision* dan *recall*, terutama ketika keduanya saling bertentangan dan tidak dapat dimaksimalkan secara bersamaan (Hicks dkk., 2022). *F1-score* menggabungkan kedua metrik tersebut menjadi gambaran keseluruhan terhadap kinerja model mengatasi ketidakseimbangan kelas. Representasi matematika untuk *F1-score* dapat dilihat pada Persamaan (43) hingga (45) (Markoulidakis dkk., 2021).

$$F1 - Score = 2 \times \frac{Precision(C_i) \times Recall(C_i)}{Precision(C_i) + Recall(C_i)} \quad (43)$$

$$F1 - Score\ (macro\ average) = 2 \times \frac{Precision(macro) \times Recall(macro)}{Precision(macro) + Recall(macro)} \quad (44)$$

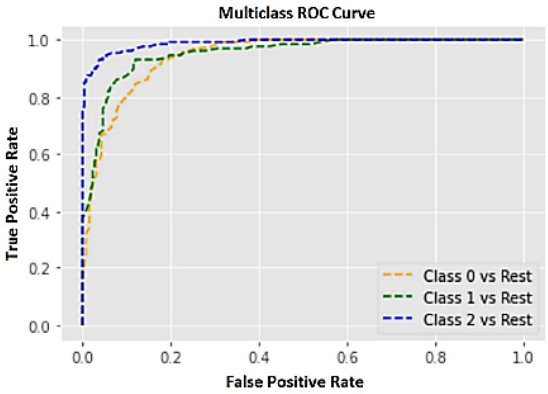
$$F1 - Score\ (micro\ average) = 2 \times \frac{Precision(micro) \times Recall(micro)}{Precision(micro) + Recall(micro)} \quad (45)$$

Pendekatan konsep Teorema Bayes untuk intepretasi *F1-score* dapat di representasikan dengan Persamaan (46).

$$F1 - Score = 2 \times \frac{Precision_{Bayesian}(C_i) \times Recall_{Bayesian}(C_i)}{Precision_{Bayesian}(C_i) + Recall_{Bayesian}(C_i)} \quad (46)$$

5. *Receiver Operating Characteristic Area Under the Curve* (ROC AUC)

Kurva ROC merupakan secara eksplisit menunjukkan keseimbangan antara TPR atau *sensitivity* dengan *False Positive Rate* (FPR) atau 1 - *specificity*. Nilai AUC dihitung sebagai area di bawah kurva ROC dan memiliki nilai antara 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan performa model yang lebih baik. Salah satu keunggulan kurva ROC adalah tidak terpengaruh oleh distribusi kelas atau biaya kesalahan (*error* cost) (Tang dkk., 2011). Nilai TPR adalah seberapa banyak model memprediksi kelas positif saat klasifikasi aktual juga positif. Nilai FPR adalah seberapa banyak model memprediksi kelas positif secara salah, saat klasifikasi aktual negatif. Kurva ROC diilustrasikan seperti pada Gambar 26.



Gambar 26. *Receiver Operating Characteristic Area Under the Curve* (Vijithananda dkk., 2023).

Berdasarkan Gambar 26, representasi matematika untuk ROC AUC dapat dilihat pada Persamaan (47) hingga (50) (Markoulidakis dkk., 2021).

$$AUC = \frac{TPR\ (C_i) - FPR(C_i) + 1}{2} = \frac{TPR\ (C_i) + TNR(C_i)}{2} \quad (47)$$

$$TPR = sensitivity\ = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (48)$$

$$TNR = specificity = \frac{TN(C_i)}{TN(C_i) + FP(C_i)} \tag{49}$$

$$AUC\ (macro\ average) = \frac{1}{N}\sum_{i=1}^{N} AUC\ (C_i) \tag{50}$$

Pendekatan konsep Teorema Bayes untuk intepretasi nilai AUC dapat di representasikan dengan Persamaan (51) hingga (54) (Jamart, 1992).

$$AUC_{Bayesian} = \frac{P(B|C_i) + P(\bar{B}|\bar{C}_i)}{2} \tag{51}$$

$$AUC_{Bayesian} = \frac{\frac{P(C_i \cap B)}{P(C_i)} + \frac{P(\bar{C}_i \cap \bar{B})}{P(\bar{C}_i)}}{2} \tag{52}$$

$$\frac{P(C_i \cap B)}{P(C_i)} = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \tag{53}$$

$$\frac{P(\bar{C}_i \cap \bar{B})}{P(\bar{C}_i)} = \frac{TN(C_i)}{TN(C_i) + FP(C_i)} \tag{54}$$

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian dilakukan pada waktu dan tempat sebagai berikut:

a. Waktu Penelitian

Penelitian dimulai pada semester ganjil tahun akademik 2024/2025. Pada bulan Agustus hingga September 2024 dengan pengumpulan bahan materi pendukung skripsi seperti buku, jurnal, dan lainnya. Selanjutnya ahnalisis data yang akan diteliti dimulai dari *Exploratory Data Analysis* (EDA) dan *preprocessing* data hingga bulan Oktober 2024. Kemudian hingga Desember 2024 deteksi label (NER-LDA), mengatasi *imbalance* data, membangun model *Transformer* (IndoBERT, DistilBERT *Multilingual*, dan XLM-RoBERTa), serta evaluasi kinerja model. Menyusun proposal penelitian dan pengajuan tema proposal penelitian pada bulan November hingga Desember 2024. Pemaparan proposal penelitian dilakukan pada bulan Januari 2025.

Integrasi perbaikan *preprocessing* data, optimalisasi model, dan evaluasi dilakukan hingga bulan Februari 2025. Penyusunan laporan hasil penelitian dilakukan pada bulan Januari hingga April 2025. Pemaparan laporan hasil penelitian dilakukan pada bulan Mei 2025. Pelaksanaan ujian komprehensif dilakukan pada bulan Mei 2025.

b. Tempat Penelitian

Penelitian ini dilakukan secara studi pustaka di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi bertempat di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedong Meneng, Bandar Lampung.

3.2 Data dan Alat Penelitian

Penelitian dilakukan dengan data dan alat sebagai berikut:

a. Data

Pada penelitian ini data yang digunakan berupa *tweet* atau opini masyarakat terkait isu kebijakan politik yang diperoleh dari https://x.com. Data dari platform X (Twitter) melalui tahap *crawling* mengintegrasikan bahasa pemrograman *Python* dengan metode *tweet harvesting* menggunakan *Application Programming Interface* (API) Twitter. *Keyword* yang digunakan adalah "Peringatan Darurat", "Darurat Kekerasan Aparat", dan "Kawal Putusan MK". Data terdiri dari dua kolom, yaitu variabel *date* dan *tweet.* Data yang digunakan terdiri dari dua bagian, yaitu data saat *keyword trending* sebanyak 12598 data dan setelah *keyword trending* sebanyak 14998 data. Fokus penelitian pada variabel *tweet* yang disimpan dalam format file *csv*. Sampel data *tweet* pada saat *keyword trending* disajikan pada Tabel 4.

Tabel 4. Sampel Data *Tweet* Saat *Keyword Trending*

	Date	Tweet
0	2024-08-21	Cerita Prabowo Gabung Kabinet Jokowi Usai Kalah …
1	2024-08-21	Selamatkan Kelas Menengah RI Ekonomi Harus …
2	2024-08-21	@ Palestine5 @jokowi Ini sudah termasuk kategori …
…	**…**	**…**
12595	2024-09-20	Sebuah pertanyaan random dari abang grab Mbak …
12596	2024-09-20	Tolong dg sangat @prabowo gak perlu koment yg …
12597	2024-09-20	@AntoniRaja @psi_id @AndiSaifulHaq Kenapa …

Data yang disajikan pada Tabel 4 merupakan kumpulan *tweet* saat *keyword trending* di X hingga selesainya rangkaian demonstrasi yang berkaitan di berbagai kota. Rentang waktu ini mencerminkan dinamika percakapan publik, mulai dari awal kemunculan isu hingga respons masyarakat selama berlangsungnya aksi demonstrasi. Sampel data *tweet* setelah *keyword trending* disajikan pada Tabel 5.

Tabel 5. Sampel Data *Tweet* Setelah *Keyword Trending*

	Date	Tweet
0	2024-10-04	Jangan diam! Suara kita adalah kekuatan kita. Mari …
1	2024-10-04	Si buyung keterima PTN dan ikut aksi #Peringatan …
2	2024-10-04	MK Menghapus Presidential Threshold tidak …
…	**…**	**…**
14995	2025-01-20	Wah...ngajak perang ini.? Belagu lu …
14996	2025-01-20	Keputusan bijak Presiden Prabowo bikin banyak …
14997	2025-01-20	Pak Prabowo membawa stabilitas mendorong …

Pada Tabel 5, terdapat kumpulan *tweet* setelah *keyword trending* di X. Data mencerminkan berbagai kebijakan publik yang diambil terkait fenomena serta dampaknya terhadap masyarakat. Selain itu, data juga mencakup berbagai permasalahan lain yang muncul sebagai konsekuensi dari peristiwa yang sebelumnya menjadi *trending*.

b. Alat

Pada penelitian ini alat yang digunakan sebagai berikut:

1. Perangkat keras

Perangkat keras yang digunakan dalam penelitian ini menjadi penunjang pelaksanaan analisis sebagai berikut:

- Processor Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz ~ 1.99 GHz
- Installed RAM 16.0 GB

2. Perangkat lunak

Perangkat lunak yang digunakan dalam penelitian ini menjadi integrasi implementasi penelitian sebagai berikut:

- Sistem Operasi Windows 10 Pro 64-bit
- *Python* 3.10.12

  Pada penelitian menggunakan Google Colaboratory sebagai platform berbasis Jupyter Notebook yang disediakan oleh Google. Adapun *package* yang digunakan dalam penelitian ini sebagai berikut:

  A. *Pandas* 2.2.2

     Pandas merupakan *package* yang berguna dalam menangani analisis data, membaca, serta menulis kembali hasil pengolahan data ke berbagai format (McKinney, 2010).

  B. *NumPy* 1.26.4

     NumPy atau *Numerical Python* merupakan *package* yang berguna dalam aljabar linier dan operasi komputasi numerik menyediakan *array* multidimensi (Harris dkk., 2020).

  C. *Scikit-learn* 1.5.2

     Scikit-learn merupakan *package* yang berguna dalam menangani data *preprocessing* serta evaluasi model (Pedregosa dkk., 2011).

D. *Matplotlib* 3.7.1

Matplotlib merupakan *package* yang berguna dalam visualisasi grafis pada data, seperti grafik, bar plot, dan lain sebagainya (Hunter, 2007).

E. *Seaborn* 0.13.2

Seaborn merupakan *package* yang berguna dalam visualisasi data secara statistik yang menarik serta informatif (Waskom, 2021).

F. *NLTK* 3.6.5

NLTK atau *Natural Language Toolkit* merupakan *package* yang berguna sebagai *toolkit* pemrosesan bahasa alami secara komprehensif (Wang dan Hu, 2021).

G. *Gensim* 4.3.3

Gensim merupakan *package* untuk implementasi LDA telah dirancang dengan korpus besar (Gupta dkk., 2022)

H. *PyLDAvis* 3.4.1

PyLDAvis merupakan *package* visualisasi LDA membantu eksplorasi hubungan antar topik dan distribusi kata (Sievert dan Shirley, 2014).

I. *PyTorch* 2.4.0

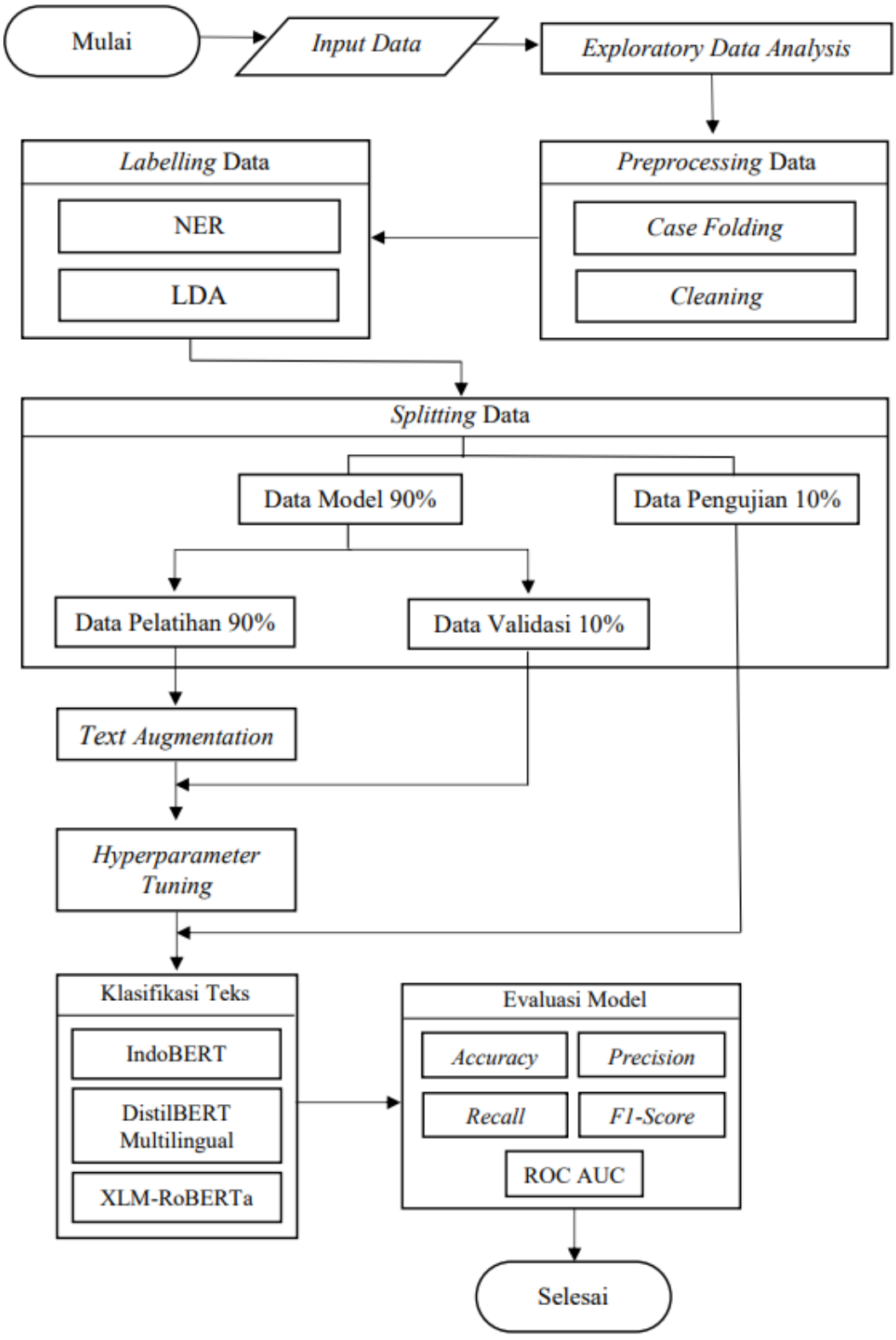
PyTorch merupakan *package* yang menyediakan akses ke berbagai model *pre-trained* seperti dalam tugas NLP model *deep learning* (Paszke dkk., 2019).

J. *Transformer* 4.44.2

Transformer merupakan *package* yang menyediakan akses implementasi *state-of-the-art* integrasi langsung dari arsitektur *Transformer* optimal dalam tugas-tugas NLP (Wolf dkk., 2020).

3.3 Alur Kerja Penelitian

Penelitian berfokus pada deteksi dan klasifikasi *tweet* isu kebijakan publik. Berdasarkan topik tersebut implementasi mengintegrasikan kemampuan berbagai algoritma dan model mencapai optimalisasi klasifikasi. Penelitian ini dilakukan berdasarkan alur penelitian pada Gambar 27.



Gambar 27. *Flowchart* Metodologi Penelitian NER-LDA berbasis Model *Transformer*.

Berdasarkan alur penelitian pada Gambar 27, berikut penjelasan setiap langkah yang dilakukan:

1. *Input* data

   Pada tahap awal, data yang digunakan dalam penelitian ini diperoleh melalui proses *crawling* platform X. Data *tweet* menjadi *input* utama untuk analisis memahami konteks kebijakan publik. Data yang digunakan pada penelitian terdiri dari dua bagian, yaitu data saat *keyword trending* sebanyak 12598 data dan setelah *keyword trending* sebanyak 14998 data.

2. *Exploratory Data Analysis* (EDA)

   EDA adalah langkah penting untuk memahami karakteristik dan struktur data sebelum mengimplementasikan algoritma matematika dan membangun model lebih lanjut. Integrasi EDA secara terstruktur dan mendalam untuk memperoleh wawasan yang berguna serta mengidentifikasi potensi masalah yang mungkin terjadi selama pemrosesan atau pembangunan model.

3. *Preprocessing* data

   Preprocessing data adalah tahap penting menyiapkan data untuk optimal diintegrasikan lebih lanjut. Beberapa tahapan *preprocessing* yang dilakukan adalah sebagai berikut:

   a. Case folding

      Case folding dalam penelitian ini merupakan proses pengubahan seluruh huruf kapital dalam teks menjadi huruf kecil untuk mengurangi kompleksitas teks dan meningkatkan konsistensi data.

   b. *Cleaning* data

      Cleaning data diimplementasikan mencakup penghapusan data duplikat, *mention*, tanda pagar, URL, serta spasi berlebih. Integrasi tersebut mengoptimalkan kualitas teks sehingga memastikan bahwa hanya teks relevan yang digunakan dalam pemrosesan model.

4. *Labelling* data

   Labelling data adalah proses memberi label pada data teks yang relevan untuk tujuan klasifikasi. Proses pelabelan ini menggabungkan beberapa teknik NLP yang canggih, seperti NER dengan model *pre-trained* IndoBERT untuk tugas NER dengan skema *tagging* IOB mengekstrak label setiap entitas atau token.

Selanjutnya, ekstraksi relasi mengintegrasikan LDA untuk mengidentifikasi 9 topik mengenai kebijakan publik, yaitu demografi, ekonomi, ideologi, kesehatan, pendidikan, pertahanan dan keamanan, politik, sosial budaya, serta sumber daya alam berdasarkan distribusi token yang berkaitan dengan label NER. Integrasi kemampuan ektraksi relasi dengan analisis mendalam menggunakan formulasi statistika memungkinkan untuk menetapkan topik kebijakan publik yang relevan secara akurat dari pola label entitas yang dihasilkan NER-LDA.

5. Pembagian data

   Pembagian data adalah langkah yang penting untuk memastikan bahwa model yang dibangun dapat diuji dengan data yang tidak terlihat selama pelatihan. Pada penelitian ini, data dibagi menjadi komposisi data model 90% dan data pengujian 10%. Selanjutnya, data model dibagi menjadi data pelatihan 90% dan data validasi 10%. Pembagian ini penting untuk menghindari *overfitting* dan memastikan bahwa model dapat digeneralisasi dengan baik pada data baru.

6. *Text augmentation*

   Ketidakseimbangan jumlah data antar kelas pada penelitian ini diatasi dengan teknik *oversampling* pada data yang memiliki label atau kategori lebih sedikit. Salah satu pendekatan yang digunakan adalah *text augmentation* menggunakan *generative* AI. Pada tahap ini, model *generative* AI *Large Language Model Meta AI* (LLaMA) 3 digunakan untuk menghasilkan data sintetik berdasarkan pola yang ada dalam data asli. *Generate* teks mengambil data teks yang sudah ada pada masing-masing label yang ingin dilakukan *oversampling* dilanjutkan dengan *prompt* ke model LLaMa diintegrasikan proses iterasi. Pada penelitian ini, *text augmentation* data hanya diterapkan pada data pelatihan. Sedangkan data validasi dan pengujian konsisten dalam distribusi aslinya. Hal ini dilakukan untuk memastikan evaluasi model mencerminkan kinerja pada data nyata tanpa intervensi yang dapat menyebabkan bias dalam pengujian.

7. *Hyperparameter tuning*

   Hyperparameter tuning adalah proses untuk menemukan kombinasi paremeter terbaik yang akan digunakan dalam model. Pada penelitian ini, *hyperparameter*

tuning mengadopsi teknik *grid search* untuk menguji parameter *batch size*, *learning rate*, dan *dropout*.

8. Membangun model klasifikasi

   Setelah memperoleh kombinasi parameter terbaik, langkah selanjutnya adalah membangun dan melatih model klasifikasi teks *multiclass* yang terdiri dari 9 label. Tiga model *Transformer* yang akan dibandingkan dalam penelitian ini adalah IndoBERT, DistilBERT *Multilingual*, dan XLM-RoBERTa. Ketiga model *Transformer* yang telah dilatih dengan korpus bahasa Indonesia dengan *framework Hugging Face*. *Fine-tuning* dilakukan dengan menggunakan data pelatihan dan validasi untuk mengoptimalkan performa model dalam tugas klasifikasi teks.

9. Evaluasi Model

   Tahap selanjutnya mengimplementasikan evaluasi performa model dengan *confusion matrix*. Berdasarkan *confusion matrix* tersebut diformulasikan metrik evaluasi yang terdiri dari *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, analisis performa model juga dilakukan dengan melihat kurva ROC AUC untuk mengukur kemampuan model dalam membedakan kelas secara menyeluruh. Hasil perhitungan seluruh metrik ini memberikan gambaran komprehensif tentang seberapa optimal model dalam mengklasifikasikan *tweet* terkait kebijakan publik. Evaluasi digunakan untuk membandingkan kinerja ketiga model *Transformer* sehingga dapat dipilih model yang memberikan hasil terbaik berdasarkan berbagai indikator evaluasi.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan penelitian yang telah terintegrasi dapat diidentifikasi kesimpulan sebagai berikut:

1. Sebaran topik kebijakan publik dalam *tweet* saat dan setelah *trending* menunjukkan adanya perbedaan pola diskusi. Fenomena ini terjadi karena label dengan topik konseptual, seperti ideologi cenderung lebih banyak dibahas saat awal tren sebagai respons terhadap isu utama yang memicu diskusi. Sementara itu, topik sosial budaya lebih banyak dibahas setelah keyword *trending* mencerminkan pergeseran diskusi menuju pembahasan yang luas terkait dampak sosial dari peristiwa tersebut.
2. Implementasi model *Transformer*, yaitu IndoBERT, DistilBERT *Multilingual*, dan XLM-RoBERTa dalam klasifikasi teks berbasis NER-LDA menunjukkan performa kinerja yang baik.
3. Pada data saat *trending*, ketiga model *Transformer* mencapai hasil terbaik dengan kombinasi parameter *batch size* 16 dan *learning rate* sebesar 1×10^{-5} dengan model DistilBERT *Multilingual* memperoleh *accuracy* tertinggi sebesar 99,21%. Pada data setelah *trending*, kombinasi parameter terbaik adalah *batch size* 32 untuk model IndoBERT serta *batch size* 16 untuk model DistilBERT *Multilingual* dan XLM-RoBERTa, sedangkan *learning rate* terbaik untuk ketiga model sebesar 1×10^{-5} dengan model IndoBERT memperoleh *accuracy* tertinggi sebesar 99,21%. Hasil menunjukkan bahwa model *Transformer* memiliki kinerja yang baik dalam klasifikasi *tweet* kebijakan publik, baik pada data saat maupun setelah *trending*.

5.2 Saran

Berdasarkan hasil dan pembahasan penelitian yang telah terintegrasi, adapun saran untuk pengembangan penelitian selanjutnya sebagai berikut:

1. Deteksi entitas dan topik dengan pendekatan NER-LDA yang dihasilkan telah memanfaatkan BERT *embedding* untuk proses *entity tagging*. Namun, analisis model lain seperti BERT-*Topic* dapat dipertimbangkan yang secara khusus dirancang menghasilkan representasi topik berdasarkan integrasi pemetaan topik berbasis BERT *embedding*.
2. Penyesuaian parameter model, seperti *batch size*, *learning rate*, *dropout*, *patience*, dan *epoch* menjadi faktor kunci dalam meningkatkan performa model klasifikasi teks. Oleh karena itu, penelitian selanjutnya dapat mengeksplorasi lebih banyak kombinasi parameter untuk memperoleh pemahaman yang lebih mendalam mencapai kinerja optimal.
3. Pertimbangan penerapan model klasifikasi teks yang optimal, seperti GPT, Gemini, dan LLaMA dapat menjadi alternatif dalam penelitian berikutnya. Perbandingan kinerja antara berbagai model dapat memberikan wawasan mendalam terkait efektivitas dan penerapannya dalam klasifikasi teks, sehingga dapat mencapai hasil yang lebih optimal sesuai dengan karakteristik data yang digunakan.

DAFTAR PUSTAKA

Adel, H., Dahou, A., Mabrouk, A., Elaziz, M. A., Kayed, M., El-Henawy, I. M., Alshathri, S., dan Ali, A. A. 2022. Improving Crisis Events Detection using Distilbert with Hunger Games Search Algorithm. *Mathematics*. **10**(3): 1-22.

Al-Behadili, H. N., Ku-Mahamud, K. R., dan Sagban, R. 2018. Rule Pruning Techniques in the Ant-Miner Classification Algorithm and Its Variants: A Review. *ResearchGate*. **1**(1): 1-7.

Alexandre, C., Gonthier, F., dan Guerra, T. 2024. What goes with what and for whom? How Political Sophistication Shapes The Structuring of Economic and Cultural Beliefs. *Investigating Forty Years of French Politics Through Value Changes*. **8**(1): 59–92.

Alim, M. S. 2024. *Ilmu Politik & Kebijakan Publik*. Sumedang: Mega Press Nusantara.

Alzubaidi, L., Zhang, J., Humaidi, A. J., AlDujaili, A., Duan, Y., AlShamma, O., Santamaria, J., Fadhel, M. A., AlAmidie, M., dan Farhan, L. 2021. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal of Big Data*. **8**(53): 1-74.

Asmara, S., dan Butsi, F. I. 2020. Twitter dan Public Sphere: Studi Fenomenologi Tentang Twitter Sebagai Media Alternatif Komunkasi Politik. *Jurnal Ilmiah Ilmu Komunikasi*. **2**(2): 75-84.

Badan Pusat Statistika. 2024. Proyeksi Penduduk Indonesia 2020-2050 Hasil Sensus Penduduk.

Basbeth, F., dan Fudholi, D. H. 2024. Klasifikasi Emosi Pada Data Text Bahasa Indonesia Menggunakan Algoritma BERT, RoBERTa, dan Distil-BERT. *Jurnal Media Informatika Budidarma*. 8(2): 1160-1170.

Berragan, C., Singleton, A., Calafiore, A., dan Morley, J. 2022. Transformer Based Named Entity Recognition for Place Name Extraction from Unstructured Text. *International Journal of Geographical Information Science*. 36(3): 1-21.

Birunda, S. S., dan Devi, R. K. 2021. A Review on Word Embedding Techniques for Text Classification. Inn*ovative Data Communication Technologies and Application*. **59**(1): 267–281.

Boban, I., Doko, A., dan Gotovac, S. 2020. Sentence Retrieval Using Stemming and Lemmatization with Different Length of the Queries. *Advances in Science, Technology and Engineering Systems Journal*. **5**(3): 349-354.

Campbell, J. C., Hindle, A., dan Stroulia, E. 2015. Latent Dirichlet Allocation: Extracting Topics From Software Engineering Data. *The Art and Science of Analyzing Software Data*. **2**(1): 139-159.

Cunha, W., Viegas, F., França, C., Rosa, T., Rocha, L., dan Gonçalves, M. A. 2023. A Comparative Survey of Instance Selection Methods Applied to Non-Neural and Transformer-Based Text Classification. *ACM Computing Surveys*. **55**(13): 1-52.

Chandradev, V., Suarjaya, I. M. A. D., dan Bayupati, I. P. A. 2023. Analisis Sentimen Review Hotel Menggunakan Metode Deep Learning BERT. *Jurnal Buana Informatika.* **14**(2): 107-116.

Chen, P. C., Tsai, H., Bhojanapalli, S., Chung, H. W., Chang, Y. W., dan Ferng, C. S. 2021. A Simple and Effective Positional Encoding for Transformers. *Cornell University*. **3**(2): 1-15.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., dan Stoyanov, V. 2019. Unsupervised Cross-Lingual Representation Learning at Scale. *Cornell University*. **2**(1): 1-12.

Cook, J., dan Ramadas, V. 2020. When to Consult Precision-Recall Curves. *The Stata Journal*. **1**(20): 131-148.

Devlin, J., Chang, M. W., Lee, K., dan Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Cornell University.* **2**(1): 1-16.

Dubey, S. R., Singh, S. K., dan Chaudhuri, B. B. 2022. Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark. *Neurocomputing*. **503**(1): 92–108.

Elgeldawi, E., Sayed, A., Galal, A. R., dan Zaki, A. M. 2021. Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*. **8**(4): 1-21.

Fanni, S. C., Febi, M., Aghakhanyan, G., dan Neri, E. 2023. Natural Language Processing. *Imaging Informatics for Healthcare Professionals: Introduction to Artificial Intelligence*. **9**(5): 87-99.

Gangadharan, V., dan Gupta, D. 2020. Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques. *Third International Conference on Computing and Network Communications (CoCoNet'19*). **171**(1): 1337–1345.

Geni, L., Yulianti, E., dan Sensuse, D. I., 2023. Sentiment Analysis of Tweets Before the 2024 Elections in Indonesia Using IndoBERT Language Models. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika.* **9**(3): 746-757.

Gillioz, A., Casas, J., Mugellini, E., dan Khaled, O. A. 2020. Overview of the Transformer-based Models for NLP Tasks. *Federated Conference on Computer Science and Information Systems.* 179-183.

Gupta, R. K., Agarwalla, R., Naik, B. H., Evuri, J. R., Thapa, A., dan Singh, T.D. 2022. Prediction of Research Trends using LDA Based Topic Modeling. *Global Transitions Proceedings*. 298-304.

Guo, Z., Wang, P., Wang, Y., dan Yu, S. 2023. Dr. LLaMA: Improving Small Language Models in Domain-Specific QA via Generative Data Augmentation. *Cornell University*. **2**(1): 1-7.

Harris, C. R., Millman, K. J., Walt, S. J. V. D., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. V., Brett, M., Haldane, A., Rio, J. F. D., Wiebe, M., Peterson, P., Marchant, P. G., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., dan Oliphant, T. E. 2020. Array Programming with NumPy. *Nature*. **585**(7825): 357–362.

Hao, Y., Dong, L., Wei, F., dan Xu, K. 2021. Self-Attention Attribution: Interpreting Information Interactions Inside Transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*. **35**(14): 12963-12971.

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., dan Yeganegi, M. R. 2020. Text Mining in Big Data Analytics. *Big Data Cognition and Computation*. **4**(1): 1-34.

Hicks, S. A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M. A., Halvorsen, P., dan Parasa, S. 2022. On Evaluation Metrics for Medical Applications of Artificial Intelligence. *Scientific Reports*. **12**(5979): 1-9.

Hidayatullah, A. F., Apong, R. A., Lai, D. T. C., dan Qazi, A. 2023. Adult Content Detection on Indonesian Tweets by Fine-Tuning Transformer-based Models. *Applied Computational Intelligence in Information Systems*. **2**(1): 1-6.

Huang, C.M. 2017. Incorporating Prior Knowledge by Selective Context Features to Enhance Topic Coherence. *International Conference on Intelligent Computing (ICIC)*. **5**(9): 310–318.

Hunter, J. D. 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. **9**(3): 90–95.

Isabona, J., Imoize, A. L., dan Kim, Y. 2022. Machine Learning-Based Boosted Regression Ensemble Combined with Hyperparameter Tuning for Optimal Adaptive Learning. *Sensors*. **22**(10): 1-22.

Jamart, J. 1992. Chance-Corrected Sensitivity and Specificity for Three-Zone Diagnostic Tests. *Journal of Clinical Epidemiology*. **45**(9): 1035-1038.

Kahloot, K. M., dan Ekler, P., 2021. Algorithmic Splitting: A Method for Dataset Preparation. *IEEE Access*. **9**(1): 125229-125237

Kandel, I., dan Castelli, M. 2020. The Effect of Batch Size on The Generalizability of The Convolutional Neural Networks on a Histopathology Dataset. *ICT Express.* **6**(4): 312-315.

Keraghel, I., Morbieu, S., dan Nadif, M. 2024. A Survey on Recent Advances in Named Entity Recognition. *Cornell University.* **4**(1): 1-30.

Khairani, U., Mutiawani, V., Ahmadian, H. 2024. Pengaruh Tahapan Preprocessing Terhadap Model Indobert dan Indobertweet untuk Mendeteksi Emosi pada Komentar Akun Berita Instagram. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **11**(4): 1-10.

Khairunnisa, S. O., Chen, Z., dan Komachi, M. 2023. Dataset Enhancement and *Multilingual* Transfer for Named Entity Recognition in the Indonesian Language. *ACM Trans. Asian Low-Resour. Lang. Inf.* **22**(6): 1-2.

Khairunnisa, S. O., Imankulova, A., dan Komachi, M. 2020. Towards a Standardized Dataset on Indonesian Named Entity Recognition. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*. **2**(1): 64–71.

Khder, M. A. 2021. Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. *International Journal of Advanced Soft Computing Applications*. **13**(3): 1–25.

Koto, F., Rahimi, A., Lau, J. H., dan Baldwin, T. 2020. IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. *COLING 2020 - The 28th International Conference on Computational Linguistics*. **4**(1): 1-14.

Kumar, A., Dabas, V., dan Hooda, P. 2020. Text Classification Algorithms for Mining Unstructured Data: a SWOT Analysis. *International Journal of Information Technology*. **12**(3): 1159–1169.

Lee, M. 2023. Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning. *Journal of Mathematics*. **2023**(1): 1-13.

Lewkowycz, A., Bahri, Y., Dyer, E., Dickstein, J. S., dan Ari, G. G. 2020. The Large Learning Rate Phase of Deep Learning: The Catapult Mechanism. *Cornell University.* 4(1): 1-28.

Li, C., Liu, H., dan Ming, Z. 2022. Empirical Study on Virtual Order of Class Labels in Nominal Classification. *International Journal of Machine Learning and Cybernetics*. **13**(1): 3255–3266.

Liu, X., dan Di, X. 2021. TanhExp: A Smooth Activation Function with High Convergencespeed for Lightweight Neural Networks. *The Institution of Engineering and Technology.* **15**(1): 136-150.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., dan Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach*. Cornell University*. **8**(1): 1-13.

Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., dan Tian, J. 2023. A Survey of Visual Transformers. *IEEE Transactions on Neural Networks and Learning Systems*. **35**(6): 1234-1245.

Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., Bian, J., dan Dou, D. 2022. Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond. *Knowledge and Information Systems.* **64**(2): 3197-3234.

Magni, P., Bellazzi, R., Nicolao, G. D., Poggesi, I., dan Rocchetti, M. 2002. Nonparametric AUC Estimation in Population Studies with Incomplete Sampling: A Bayesian Approach. *Journal of Pharmacokinetics and Pharmacodynamics*. **29**(1): 445-471.

Markoulidakis, I., Kopsiaftis, G., Rallis, I., dan Georgoulas, I. 2021. Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*. **9**(4): 1-22.

McKinney, W. 2010. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*. 51–56.

Mohammed, H. H., Dogdu, E., Görür, A. K., dan Choupani, R. 2020. Multi-Label Classification of Text Documents Using Deep Learning. *2020 IEEE International Conference on Big Data*. **8**(1): 4681-4689.

Nabiilah, G. Z., Prasetyo, S. Y., Izdihar, Z. N., dan Girsang, A. S. 2023. BERT Base Model for Toxic Comment Analysis on Indonesian Social Media. *7th International Conference on Computer Science and Computational Intelligence.* **216**(1): 714-721.

Nallapati, R., Surdeanu, M., dan Manning, C. 2010. Blind Domain Transfer for Named Entity Recognition Using Generative Latent Topic Models. *Proceedings of the NIPS 2010 Workshop on Transfer Learning Via Rich Generative Models*. **1**(1): 281–289.

Nandanwar, A. K., dan Choudhary, J. 2023. Contextual Embeddings-Based Web Page Categorization Using the Fine-Tune BERT Model. *Symmetry*. **15**(2). 1-17.

Nourbakhsh, Z., dan Habibi, N. 2022. Combining LSTM and CNN Methods and Fundamental Analysis for Stock Price Trend Prediction. *Multimedia Tools and Applications.* **82**(1): 17769-17799.

Nurdin, A., Aji, B. A. S., Bustamin, A., dan Abidin, Z. 2020. Perbandingan Kinerja Word Embedding Word2Vec, Glove, dan Fasttext Pada Klasifikasi Teks. *Jurnal TEKNOKOMPAK.* **14**(2): 74-79.

Obla, S., Gong, X., Aloufi, A., Hu, P., dan Takabi, D. 2020. Effective Activation Functions for Homomorphic Evaluation of Deep Neural Networks. *IEEE Access.* **8**(1): 153098-153112.

Ouellette, S.. A., dan Rudzicz, F. 2020. On Losses for Modern Language Models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 3(1): 403-413.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., dan Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library *Advances in Neural Information Processing Systems*. **32**(1): 1-12.

Patil, N., Patil, A., dan Pawar, B. V. 2020. Named Entity Recognition using Conditional Random Fields. *Procedia Computer Science*. **167**(1): 1181–1188.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perror, M., dan Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. **12**(1): 2825–2830.

Priyanka, P., Kalyani, S., dan Chatterjee, A. 2024. Learning Rate Optimization for Deep Neural Networks Using Lipschitz Bandits. *Cornell University*. **2**(1): 1-5.

Qamar, U., dan Raza, M. S. 2020. Text Mining. *Data Science Concepts and Techniques with Applications*. **5**(2): 133–151.

Qibtiyah, D. B. 2024. Daya Tarik Live Streaming TikTok Shop: Bagaimana Interaksi Langsung Mempengaruhi Niat Pembelian Produk Fashion. *EKOMA: Jurnal Ekonomi, Manajemen, Akuntansi*. **3**(5): 1-14.

Ramadhan, M. F. A., Setiawan, I. R., dan Asriyanik, A. 2024. Klasifikasi Hoax dan Fakta Menggunakan Algoritma Shallow Neural Network Pada Berita Politik Pemilihan Presiden Indonesia 2024. *Jurnal Mahasiswa Teknik Informatika.* **8**(4): 1-8.

Ramshaw, L., dan Marcus, M. 1995. Text Chunking using Transformation-Based Learning. *Third Workshop on Very Large Corpora*. **21**(1): 82-94.

Ravichandiran, S. 2021. *Getting Started with Google BERT: Build and Train State-Of-The-Art Natural Language Processing Models using BERT.* Packt Publishing.

Roodschild, M., Sardinas, J. G., Jimenez, V. A., dan Will, A. 2024. Zorro: A Flexible and Differentiable Parametric Family of Activation Functions That Extends ReLU and GELU. *Neural and Evolutionary Computing*. **8**(2): 1-13

Sanh, V., Debut, L., Chaumond, J., dan Wolf, T. 2020. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *Cornell University*. **4**(1): 1-5.

Saunshi, N., Gupta, A., dan Hu, W. 2021. A Representation Learning Perspective on the Importance of Train-Validation Splitting in Meta-Learning. *Proceedings of the 38th International Conference on Machine Learning.* **38**(5): 1-11.

Setijohatmo, U. T., Rachmat, S., Susilawati, T., dan Rahman, Y. 2020. Analisis Metode Latent Dirichlet Allocation Klasifikasi Dokumen Laporan Tugas Akhir Berdasarkan Pemodelan Topik. *Prosiding The 11th Industrial Research Workshop and National Seminar*. **11**(1): 402-408

Shah, K., Patel, H., Sanghvi, D., dan Shah, M. 2020. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augmented Human Research*. **5**(12): 1-16.

Shen, S. L., Zhang, N., Zhou, A., dan Yin, Z. Y. 2022. Enhancement of Neural Networks with an Alternative Activation Function TanhLU. *Expert Systems with Applications*. **199**(5): 117181-117192.

Shin, S., Lee, Y., Kim, M., Park, J., Lee, S., dan Min, K. 2020. Deep Neural Network Model with Bayesian Hyperparameter Optimization for Prediction of NO at Transient Conditions in a Diesel Engine. *Engineering Applications of Artificial Intelligence*. **94**(1): 1:14.

Shorten, C., Khoshgoftaar, T. M., dan Furht, B. 2021. Text Data Augmentation for Deep Learning. *Journal of Big Data*. **8**(101): 1-34.

Sievert, C., dan Shirley, K. E. 2014. LDAvis: A Method for Visualizing and Interpreting Topics. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. 63–70.

Soydaner, D. 2020. A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*. **34**(13): 1-23.

Statista Research Department. 2024. X (formerly Twitter) - statistics & facts.

Syed, S., dan Spruit, M. 2017. Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation. *International Conference on Data Science and Advanced Analytics.* **2**(1): 165-174.

Syukron, A., dan Subekti, A. 2018. Penerapan Metode Random Over-Under Sampling dan Random Forest untuk Klasifikasi Penilaian Kredit. *Jurnal Informatika*. **5**(2): 175-185.

Tanaka, H., Shinnou, H., Cao, R., Bai, J., dan Ma, J. 2020. Document Classification by Word Embeddings of BERT. *Computational Linguistics*. **1215**(1): 145-154.

Tang, K., Wang, R., dan Chen, T. 2011. Towards Maximizing the Area Under the ROC Curve for Multi-Class Classification Problems. *Proceedings of the AAAI Conference on Artificial Intelligence*. **1**(25): 483-488.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., dan Polosukhin, I. 2017. Attention Is All You Need. *31st Conference on Neural Information Processing Systems*. **22**(5): 6000-6010.

Vijithananda, S. M., Jayatilake, M. L., Goncalves, T. C., Rato, L. M., Weerakoon, B. S., Kalupahana, T. D., Silva, A. D., Dissanayake, K., dan Hewavithana, P. B. 2023. Texture Feature Analysis of MRI-ADC Images to Differentiate Glioma Grades Using Machine Learning Techniques. *Scientific Reports*. **13**(15772): 1-15.

Wang, M., dan Hu, F. 2021. The Application of NLTK Library for Python Natural Language Processing in Corpus Research. *Theory and Practice in Language Studies*. **11**(9): 1041-1049.

Waskom, M. L. 2021. Seaborn: Statistical Data Visualization. *Journal of Open Source Software*. **6**(60): 1-4.

We Are Social. 2024. Digital 2024: 5 Bilion Social Media Users - We Are Social Indonesia.

Widyawan., Utomo, B. P., dan Rizal, M. N. 2024. A Novel Fusion of Machine Learning Methods for Enhancing Named Entity Recognition in Indonesian Language Text. *Jurnal Sistem Informasi Bisnis*. **14**(4): 311-320.

Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., Li, X., Lim, Z. Y., Soleman, S., Mahendra, R., Fung, P., Bahar, S., dan Purwarianti, A. 2020. IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. **15**(2): 843–857.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P. V., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., dan Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* 38–45.

Yi, D., Ahn, J., dan Ji, S. 2020. An Effective Optimization Method for Machine Learning Based on ADAM. *Applied Sciences*. **10**(3): 1-20.

Younisse, R., Ghnemat, R., dan Alsaraireh, J. 2023. Fine-tuning U-net for Medical Image Segmentation Based on Activation Function, Optimizer and Pooling Layer. *International Journal of Electrical and Computer Engineering.* **13**(5): 5406-5417.

Yuyun, W., Latief, A. D., Sampurno, T., Zainuddin, H., Arisha, A. O., dan Mushaf. 2023. Next Sentence Prediction: The Impact of Preprocessing Techniques in Deep Learning. *International Conference on Computer, Control, Informatics and its Applications*. **11**(9): 274-278.