

**ANALISIS PERFORMA *TRANSFORMER-BASED MODEL* PADA  
DETEKSI *BLACK CAMPAIGN*  
(Studi Kasus: Pemilihan Presiden Indonesia 2024)**

**(Skripsi)**

**Oleh**

**JESSEN RAMADEKSA ALLEN  
NPM. 2217051070**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

**ANALISIS PERFORMA *TRANSFORMER-BASED MODEL* PADA  
DETEKSI *BLACK CAMPAIGN*  
(Studi Kasus: Pemilihan Presiden Indonesia 2024)**

Oleh

**JESSEN RAMADEKSA ALLEN**

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

## ABSTRAK

### ANALISIS PERFORMA *TRANSFORMER-BASED MODEL* PADA DETEKSI *BLACK CAMPAIGN* (Studi Kasus: Pemilihan Presiden Indonesia 2024)

Oleh

JESSEN RAMADEKSA ALLEN

Meningkatnya penggunaan media sosial pada Pemilihan Presiden Indonesia 2024 memperluas ruang komunikasi politik sekaligus memunculkan praktik *black campaign* yang berpotensi merusak reputasi kandidat dan menyesatkan opini publik. Pendekatan klasifikasi teks yang digunakan pada studi awal masih memiliki keterbatasan dalam menangkap hubungan semantik yang kompleks pada teks politik, sehingga diperlukan metode yang mampu memahami representasi bahasa secara lebih mendalam. Penelitian ini bertujuan untuk menganalisis performa model berbasis *Transformer* dalam mendeteksi konten *black campaign* pada data Twitter terkait Pemilihan Presiden Indonesia 2024. Dataset yang digunakan berjumlah 7.025 data dengan dua kelas, yaitu *Black Campaign* dan *Non-Black Campaign*.

Tahapan penelitian meliputi *preprocessing* teks, penyeimbangan data menggunakan *oversampling* dan *downsampling*, *splitting*, proses *hyperparameter tuning* dan *fine-tuning*, serta evaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Model yang diuji meliputi *IndoBERT*, *IndoBERTweet*, dan *XLM-RoBERTa*. Hasil penelitian menunjukkan bahwa seluruh model *Transformer* mampu melampaui performa model *benchmark*, dengan performa terbaik diperoleh oleh *IndoBERT* menggunakan pendekatan *oversampling* dan normalisasi teks yang mencapai akurasi 95,45%. Pengujian pada dataset eksternal menunjukkan kemampuan model dalam menghadapi variasi distribusi data dengan akurasi sekitar 50%. Secara keseluruhan, arsitektur *Transformer* menunjukkan kemampuan yang lebih baik dalam memahami representasi semantik teks politik dibandingkan model sekuensial.

**Kata Kunci:** *Black Campaign*, *Transformer-based Model*, Klasifikasi Teks, *Natural Language Processing*, Media Sosial, Pemilihan Presiden 2024, *IndoBERT*.

## ABSTRACT

### PERFORMANCE ANALYSIS OF TRANSFORMER-BASED MODELS IN BLACK CAMPAIGN DETECTION (Case Study: The 2024 Indonesian Presidential Election)

By

JESSEN RAMADEKSA ALLEN

The increasing use of social media during the 2024 Indonesian Presidential Election has expanded the space for political communication while also enabling the spread of black campaign content that may damage candidates reputations and mislead public opinion. Early text classification approaches still face limitations in capturing complex semantic relationships within political texts, highlighting the need for methods capable of learning deeper language representations. This study aims to analyze the performance of Transformer-based models in detecting black campaign content on Twitter data related to the 2024 Indonesian Presidential Election. The dataset used consists of 7,025 labeled instances with two classes: Black Campaign and Non-Black Campaign.

The research stages include text preprocessing, data balancing using oversampling and downsampling, data splitting, hyperparameter tuning, and fine-tuning processes, followed by evaluation using Accuracy, Precision, Recall, and F1-score metrics. The models evaluated in this study include IndoBERT, IndoBERTweet, and XLM-RoBERTa. The results show that all Transformer models outperform the benchmark model, with the best performance achieved by IndoBERT using oversampling and text normalization, reaching an accuracy of 95.45%. Evaluation on an external dataset further provides insight into the model's ability to handle variations in data distribution, achieving an accuracy of approximately 50%. Overall, the Transformer architecture demonstrates a stronger capability in capturing the semantic representation of political texts compared to sequential models.

**Keywords:** Black Campaign, Transformer-based Model, Text Classification, Natural Language Processing, Social Media, 2024 Presidential Election, IndoBERT.

Judul Skripsi : **ANALISIS PERFORMA *TRANSFORMER-BASED MODEL* PADA DETEKSI *BLACK CAMPAIGN* (Studi Kasus: Pemilihan Presiden Indonesia 2024)**

Nama Mahasiswa : **Jessen Ramadeksa Allen**

Nomor Pokok Mahasiswa : 2217051070

Program Studi : S1 Ilmu Komputer

Fakultas : Fakultas Matematika dan Ilmu Pengetahuan Alam



**MENYETUJUI**

1. Komisi Pembimbing

**Rahman Taufik, S.Pd., M. Kom.**  
NIP. 19930627 202203 1 007

**Dhella Amelia, M.Kom.**  
NIP. 19900127 202406 2 001

2. Mengetahui

Ketua Jurusan Ilmu Komputer

Ketua Program Studi

**Dwi Sakethi, S.Si., M.Kom.**  
NIP. 19680611 199802 1 001

**Tristiyanto, M.I.S., Ph.D.**  
NIP. 19810414 200501 1 001

**MENGESAHKAN**

1. Tim Penguji

Ketua : **Rahman Taufik, S.Pd., M. Kom.**



.....

Sekretaris : **Dhella Amelia, M.Kom.**



.....

Penguji Utama : **Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc.**



.....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng. Heri Satria, S.Si., M.Si.**  
NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: **6 Maret 2026**

## PERNYATAAN

Saya bertanda tangan di bawah ini:

Nama : Jessen Ramadeksa Allen

NPM : 2217051070

Dengan ini menyatakan bahwa skripsi saya yang berjudul "**ANALISIS PERFORMA TRANSFORMER-BASED MODEL PADA DETEKSI BLACK CAMPAIGN (Studi Kasus: Pemilihan Presiden Indonesia 2024)**" merupakan hasil karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 2 April 2026  
Penulis,



Jessen Ramadeksa Allen  
NPM. 2217051070

## RIWAYAT HIDUP



Penulis bernama Jessen Ramadeksa Allen, lahir di Pagar Alam pada tanggal 13 Juni 2005. Penulis merupakan anak keempat dari empat bersaudara. Pendidikan dasar diselesaikan di SD Negeri 70 Pagar Alam pada Tahun 2016, kemudian melanjutkan pendidikan di SMP Xaverius Pagar Alam dan lulus pada Tahun 2019. Pendidikan menengah atas ditempuh di SMA Unggul Negeri 1 Pagar Alam dan diselesaikan pada Tahun 2022. Pada tahun yang sama, penulis diterima sebagai mahasiswa Program Studi S1 Ilmu Komputer Universitas Lampung melalui jalur SBMPTN.

Selama menempuh pendidikan di Program Studi S1 Ilmu Komputer, penulis aktif dalam berbagai kegiatan, antara lain sebagai berikut:

1. *Brand Ambassador* UKM-U English Society Unila (ESo) Tahun 2023.
2. *Staff Public Relations* UKM-U English Society Unila (ESo) Tahun 2024.
3. *Content Creator* UKM-U English Society Unila (ESo) Tahun 2024.
4. *Copywriter* UKM-U English Society Unila (ESo) Tahun 2024
5. Kuliah Kerja Nyata (KKN) Periode I Tahun 2025 di Desa Tulung Kakan, Kecamatan Bumi Ratu Nuban, Kabupaten Lampung Tengah.
6. Koordinator Asisten Dosen Mata Kuliah Analisis dan Desain Sistem Informasi Tahun 2025.
7. Kerja Praktik (KP) di PT Bukit Asam Tbk, Tanjung Enim, Sumatera Selatan Tahun 2025.
8. Juara 3 Lomba Baca Puisi Putra pada Pekan Seni Mahasiswa Universitas Lampung (PEKSIMILA) Tahun 2025.

## MOTTO

*“Allah doesn’t burden any soul beyond its capacity.”*

(Q.S Al-Baqarah:286)

*“People who push themselves forward see a different kind of scenery.”*

(Eren Yeager)

*“Everything will be fine, even if it’s a long way round.”*

(Wave to Earth)

*“How can we judge people for their choices when we know nothing about their option.”*

## **PERSEMBAHAN**

Segala puji saya panjatkan kepada Allah Subhanahu wa Ta'ala atas segala rahmat, karunia dan hidayah-Nya. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad Shallallahu'alaihi wa sallam sehingga saya dapat menyelesaikan skripsi ini dengan baik.

### **Skripsi ini saya persembahkan kepada:**

Kedua orang tua saya, keluarga serta seluruh rekan yang telah senantiasa kebersamai dan membantu dalam proses penyelesaian skripsi ini. Kemudian untuk diri saya sendiri, terima kasih atas keberhasilan dalam menuntaskan pendidikan di Program Studi S1 Ilmu Komputer Universitas Lampung.

## SANWACANA

Segala puji bagi Allah Subhanahu wa Ta'ala yang telah melimpahkan karunia-Nya sehingga penulis berhasil menuntaskan skripsi ini sesuai jadwal yang ditentukan. Shalawat serta salam semoga selalu terlimpahkan kepada Nabi Muhammad Shallallahu 'alaihi wa sallam.

Karya ilmiah berjudul "**ANALISIS PERFORMA *TRANSFORMER-BASED MODEL* PADA DETEKSI *BLACK CAMPAIGN* (Studi Kasus: Pemilihan Presiden Indonesia 2024)**" ini merupakan prasyarat akademis untuk meraih gelar sarjana di Program Studi S1 Ilmu Komputer Universitas Lampung.

Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penyusunan skripsi ini yaitu:

1. Kedua orang tua tercinta, Papa dan Mama sebagai pilar kekuatan utama penulis. Terima kasih atas doa yang tidak pernah putus, kasih sayang yang tulus serta pengorbanan luar biasa yang telah diberikan hingga penulis dapat meraih gelar sarjana ini.
2. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
3. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
4. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Lampung dan dosen Pembimbing Akademik.

6. Bapak Rahman Taufik, S.Pd., M.Kom. selaku dosen Pembimbing Utama yang telah memberikan arahan, ide, kritik dan saran dalam menyelesaikan penelitian ini.
7. Ibu Dhella Amelia, M.Kom. selaku dosen Pembimbing Kedua yang selalu meluangkan waktunya untuk membimbing, memberikan arahan, motivasi dan bantuan ketika mengalami kesulitan dalam menyelesaikan penelitian ini.
8. Bapak Dr. rer. nat. Akmal Junaidi, S.Si., M.Sc. selaku dosen Pembahas yang telah memberikan kritik dan saran yang bermanfaat dalam perbaikan skripsi.
9. Bapak, Ibu Dosen dan semua Staf Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan banyak ilmu dan membantu segala urusan administrasi penulis.
10. Bapak Wisnu Priambodo dan Bapak Eri Zuliarso selaku peneliti terdahulu yang telah memberikan izin kepada penulis untuk menggunakan dataset dalam penelitian ini.
11. Keluarga besar penulis Allen Family yaitu Mbak, Teteh, Kak Yen, Abang, Dedek dan Adek yang selalu menjadi alasan bagi penulis untuk tetap bertahan dan bersemangat. Terima kasih atas dukungan moril, materiil, serta kepercayaan luar biasa yang telah diberikan selama masa perkuliahan hingga selesainya tugas akhir ini.
12. Rekan-rekan The Purnama's yang telah menjadi rekan berbagi cerita dan tawa yang luar biasa. Kehadiran yang membuat dinamika perkuliahan di Ilmu Komputer menjadi jauh lebih berkesan dan penuh warna.
13. Rekan-rekan di Yapping yang telah memberikan berbagai diskusi, obrolan ringan, serta sesi berbagi pikiran yang senantiasa menjadi penyemangat dan penghilang penat di tengah proses penyusunan skripsi ini.
14. Teman-teman Suka Homesick yang telah menjadi tempat berbagi cerita, berkeluh kesah, serta saling menguatkan ketika rasa rindu akan rumah datang di tengah perjuangan menyelesaikan studi di perantauan.

15. Rekan-rekan KKN Desa Tulung Kakan terima kasih atas kebersamaan, pelajaran hidup, dan kenangan indah yang telah kita ciptakan bersamamelama masa pengabdian.
16. Keluarga besar Ilmu Komputer 2022 yang telah berjuang bersama selama perkuliahan.
17. Teman-teman Kos Pondok Indah yang telah memberikan dukungan selama proses perkuliahan penulis
18. Ahlan dan Firmansyah sahabat sekaligus saudara dalam perjuangan. Terima kasih telah menjadi saksi setiap fase senang, sedih hingga masa-masa sulit penulis. Terima kasih telah tumbuh dan berkembang bersama serta selalu memberikan dukungan tanpa henti.
19. Terakhir, apresiasi setinggi-tingginya penulis tujukan kepada diri penulis sendiri. Terima kasih atas segala dedikasi, ketangguhan dan konsistensi yang telah dikerahkan dalam menghadapi berbagai tantangan akademik selama masa studi. Terima kasih telah mampu menyelesaikan apa yang telah dimulai dengan penuh tanggung jawab hingga penulis sampai pada titik pencapaian ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna karena keterbatasan pengetahuan, pengalaman, dan kemampuan penulis. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat dan kontribusi bagi pengembangan ilmu pengetahuan khususnya di bidang Ilmu Komputer.

Bandar Lampung, 2 April 2026  
Penulis,



Jessen Ramadexsa Allen  
NPM. 2217051070

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI</b> .....	<b>v</b>
<b>DAFTAR GAMBAR</b> .....	<b>viii</b>
<b>DAFTAR TABEL</b> .....	<b>ix</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Penelitian Terdahulu.....	5
2.2 <i>Black Campaign</i> .....	9
2.3 <i>Natural Language Processing</i> .....	11
2.4 <i>Supervised Learning</i> .....	11
2.5 <i>Transformer-based Model</i> .....	11
2.5.1 <i>Self-Attention</i> .....	14
2.5.2 <i>Encoder Transformer</i> .....	15
2.5.3 <i>Decoder Transformer</i> .....	16
2.5.4 <i>Multi-Head Attention</i> .....	17
2.5.5 <i>Feed Forward</i> .....	19
2.6 Model berbasis <i>Transformer</i> .....	20
2.6.1 <i>IndoBERT</i> .....	20
2.6.2 <i>IndoBERTweet</i> .....	21

2.6.3 <i>XLNet</i> .....	22
2.7 <i>Pre-processing</i> .....	23
2.7.1 <i>Data Cleaning</i> .....	23
2.7.2 <i>Case Folding</i> .....	23
2.7.3 <i>Normalisasi Teks</i> .....	24
2.7.4 <i>Oversampling</i> .....	24
2.7.5 <i>Downsampling</i> .....	24
2.8 <i>Splitting Data</i> .....	25
2.8.1 <i>Data Train</i> .....	25
2.8.2 <i>Data Test</i> .....	25
2.8.3 <i>Data Validation</i> .....	26
2.9 <i>Tokenisasi</i> .....	26
2.9.1 <i>BertTokenizer</i> .....	26
2.9.2 <i>XLNetTokenizer</i> .....	27
2.10 <i>Fine-Tuning</i> .....	27
2.10.1 <i>Hyperparameter Tuning</i> .....	28
2.11 <i>Hyperparameter</i> .....	28
2.11.1 <i>Epoch</i> .....	29
2.11.2 <i>Batch Size</i> .....	29
2.11.3 <i>Learning Rate</i> .....	29
2.11.4 <i>Early Stopping</i> .....	29
2.11.5 <i>Weight Decay</i> .....	30
2.12 <i>Overfitting</i> .....	31
2.13 <i>Underfitting</i> .....	31
2.14 <i>Confusion Matrix</i> .....	32
2.14.1 <i>Accuracy</i> .....	33
2.14.2 <i>Precision</i> .....	33
2.14.3 <i>Recall</i> .....	33
2.14.4 <i>F1-Score</i> .....	34
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>35</b>
3.1 <i>Tempat dan Waktu</i> .....	35
3.1.1 <i>Tempat</i> .....	35

3.1.2 Waktu .....	35
3.2 Perangkat Penelitian.....	35
3.2.1 Perangkat Keras .....	35
3.2.2 Perangkat Lunak .....	36
3.3 Dataset.....	36
3.4 Tahapan Penelitian .....	37
3.4.1 Pengumpulan Data .....	39
3.4.2 <i>Pre-processing</i> .....	39
3.4.3 <i>Splitting</i> Data .....	40
3.4.4 Tokenisasi.....	41
3.4.5 Model .....	42
3.4.6 <i>Fine-Tuning</i> .....	43
3.4.7 Evaluasi Model .....	45
3.4.8 Pengujian Generalisasi Model .....	46
3.4.9 Analisis Performa.....	47
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>48</b>
5.1 Kesimpulan .....	48
5.2 Saran .....	49
<b>DAFTAR PUSTAKA .....</b>	<b>50</b>

## DAFTAR GAMBAR

Gambar	Halaman
1. Gambar 1 Arsitektur Model <i>Transformer</i> (Vaswani, 2017).....	13
2. Gambar 2 Ilustrasi <i>Query Key Value</i> (Nyandwi, 2023) .....	15
3. Gambar 3 Ilustrasi <i>Multi-Head Attention</i> .....	18
4. Gambar 4 Ilustrasi <i>Scaled Dot-product Attention</i> (Nyandwi, 2023).....	19
5. Gambar 5 Ilustrasi <i>Feed Forward</i> (Nyandwi, 2023) .....	20
6. Gambar 6 Arsitektur <i>IndoBERT</i> (Yulianti dan Nissa, 2024).....	21
7. Gambar 7 Arsitektur <i>XLM-RoBERTa</i> (Ramanathan et al., 2023) .....	22
8. Gambar 8 Ilustrasi <i>Overfitting</i> (Aliferis and Simon, 2024) .....	31
9. Gambar 9 Ilustrasi <i>Underfitting</i> (Aliferis and Simon, 2024).....	32
10. Gambar 10 Tahapan Penelitian .....	38

## DAFTAR TABEL

Tabel	Halaman
1. Tabel 1 Penelitian Terdahulu.....	5
2. Tabel 2 <i>Confusion Matrix</i> (Maxwell and Warner, 2021) .....	32
3. Tabel 3 Contoh Data Penelitian .....	36

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Transformasi digital telah mengubah lanskap komunikasi politik secara fundamental. Media sosial saat ini telah menjadi wadah utama bagi para kandidat politik untuk berinteraksi dengan publik dan menyebarluaskan kegiatan politik mereka. Dalam konteks demokrasi Indonesia, khususnya pada Pemilihan Umum 2024, media sosial memainkan peran krusial dalam membentuk persepsi publik terhadap para kandidat. Namun, penggunaan media sosial di ranah politik disertai dengan tantangan signifikan yang dapat mengancam integritas proses demokrasi, yaitu meningkatnya fenomena kampanye hitam (Jeconiah, 2025).

Menurut Azhari dan Muhjad (2024), kampanye hitam di media sosial telah menjadi fenomena yang meresahkan masyarakat. Praktik ini didefinisikan sebagai upaya menyebarkan isu atau gosip yang ditujukan kepada pihak lawan tanpa didukung oleh fakta atau bukti yang jelas, sehingga cenderung mengarah pada fitnah Azhari dan Muhjad (2024). Penyebaran narasi yang belum teruji kebenarannya tersebut tidak hanya berpotensi merusak reputasi kandidat tetapi juga dapat menyebabkan opini publik yang menyesatkan Jeconiah (2025). Sulitnya penuntutan berbagai kasus kampanye hitam di ruang digital menekankan urgensi pengembangan solusi teknologi untuk mendeteksi konten tersebut secara otomatis (Azhari dan Muhjad, 2024).

Menjawab tantangan tersebut, penelitian sebelumnya oleh Priambodo dan Zuliarso (2024) telah meletakkan fondasi penting melalui pengembangan sebuah model deteksi kampanye hitam. Dalam penelitian tersebut, algoritma *K-Means* diaplikasikan untuk melakukan pengelompokan data teks. Proses ini memerlukan

intervensi manusia untuk menentukan label yang tepat pada setiap kelompok data (*expert judgment*), sehingga menghasilkan sebuah himpunan data (dataset) berlabel yang menjadi *ground truth*. Himpunan data ini selanjutnya digunakan untuk melatih model klasifikasi *Long Short-Term Memory* (LSTM) dan berhasil menetapkan kinerja dasar (*baseline*) dengan capaian akurasi sebesar 90.28% (Priambodo dan Zuliarso, 2024).

Keberhasilan penelitian terdahulu dalam menghasilkan himpunan data berlabel berkualitas tinggi serta menetapkan sebuah tolok ukur kinerja membuka peluang untuk analisis yang lebih mendalam. Dengan ketersediaan himpunan data tersebut, langkah logis berikutnya adalah membangun dan menganalisis model klasifikasi otomatis yang andal. Penelitian oleh Priambodo dan Zuliarso (2024) telah membuktikan efektivitas arsitektur berbasis sekuensial *Long Short-Term Memory* (LSTM) yang memiliki keunggulan dalam memahami urutan kata. Akan tetapi, muncul permasalahan khusus terkait teknik tokenisasi yang digunakan pada penelitian terdahulu, yaitu *Term Frequency–Inverse Document Frequency* (TF-IDF). Metode ini bekerja dengan prinsip *Bag-of-Words* (BoW) yang sekadar menghitung seberapa sering suatu kata muncul dalam dokumen di mana sekumpulan kata yang muncul dalam sebuah dokumen tanpa memperhatikan urutan maupun maknanya, di mana setiap kata hanya dilihat berdasarkan seberapa sering kata tersebut muncul. Namun, TF-IDF memiliki keterbatasan mendasar karena tidak dapat memahami konteks semantik, menangani ambiguitas makna, maupun merepresentasikan hubungan antar kata dalam kalimat. Akibatnya, representasi teks yang dihasilkan menjadi kurang kuat dalam menghadapi karakteristik bahasa informal di media sosial yang cenderung fleksibel, seperti penggunaan *slang*, akronim, dan susunan kalimat yang tidak terstruktur. Keterbatasan tersebut berdampak pada menurunnya kinerja model klasifikasi ketika digunakan untuk menganalisis data teks dari platform seperti Twitter (Widhi, 2025).

Untuk mengatasi keterbatasan kontekstual pada pendekatan TF-IDF, penelitian ini menggunakan arsitektur *Transformer-based model*. Tidak seperti metode *bag-of-words*, *Transformer-based model* mampu menangkap makna kata secara mendalam melalui mekanisme *self-attention* yang menganalisis keterkaitan setiap kata dengan

seluruh kata lain dalam kalimat secara dua arah. Proses ini menghasilkan *contextual embedding*, yaitu representasi numerik kata yang menyesuaikan dengan konteks pemakaiannya. Selain itu, *tokenizer* berbasis *sub-word* seperti *WordPiece* atau *Byte-Pair Encoding* (BPE) memungkinkan pemrosesan kata-kata di luar kosakata umum yang sering muncul pada teks media sosial dengan memecahnya menjadi unit sub-kata yang lebih kecil. Pendekatan ini menghasilkan representasi vektor yang lebih presisi dan tangguh untuk mendukung performa klasifikasi teks (Komaran, 2025).

Kondisi ini memunculkan sebuah celah analisis (*analytical gap*): “Bagaimana penerapan arsitektur modern berbasis *Transformer-based model* ketika diterapkan pada dataset dan tugas klasifikasi yang identik seperti *black campaign*?”. Oleh karena itu, penelitian ini diusulkan untuk menganalisis secara performa dari arsitektur *Transformer-based model* dalam menangani kompleksitas linguistik pada *Black Campaign* di Indonesia. Fokus utama penelitian adalah untuk mengkaji dan mengukur kinerja arsitektur modern ini menggunakan metrik evaluasi standar. Untuk memberikan konteks dan menginterpretasikan signifikansi dari hasil yang diperoleh, performa dari *Transformer-based model* ini akan disandingkan dengan tolok ukur (*benchmark*) yang telah ditetapkan oleh arsitektur sekuensial *Long Short-Term Memory* (LSTM) dari penelitian sebelumnya. Penelitian ini diharapkan dapat menyumbangkan wawasan yang berharga mengenai kemampuan dan perilaku *Transformer-based model* untuk tugas klasifikasi teks politik dalam konteks kebahasaan Indonesia.

## 1.2 Rumusan Masalah

Rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan *Transformer-based model* dalam mengklasifikasikan *Black Campaign* pada dataset Pemilihan Presiden Indonesia 2024.
2. Bagaimana kinerja arsitektur *Transformer-based model* mengklasifikasikan *black campaign* pada studi kasus Pemilihan Presiden Indonesia 2024.

### 1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian ini menggunakan dataset dari penelitian Priambodo dan Zuliarso (2024) yang berisi data teks Twitter dengan 2 kelas, yaitu *Black Campaign* dan *Non-Black Campaign*.
2. Penelitian ini terbatas pada analisis kinerja model *Transformer-based model* dalam mengklasifikasikan *black campaign*.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menerapkan arsitektur *Transformer-based model* untuk melakukan klasifikasi *black campaign*.
2. Mengukur performa *Transformer-based model* dalam klasifikasi *black campaign* menggunakan metrik evaluasi standar, meliputi akurasi, presisi, *recall*, dan *F1-score*.
3. Menganalisis kinerja *Transformer-based model* dengan menjadikannya sebagai perbandingan (*benchmark*) performa yang telah dicapai oleh penelitian sebelumnya.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu.

1. Menambah wawasan keilmuan mengenai karakteristik arsitektur *Transformer-based model* dalam klasifikasi *black campaign*.
2. Menyediakan landasan dasar faktual yang dapat digunakan sebagai acuan dalam pengembangan sistem deteksi konten politik negatif berbasis kecerdasan buatan di masa mendatang.
3. Menjadi dasar perbandingan (*benchmark*) yang bermanfaat bagi peneliti lain, sehingga dapat membuat keputusan yang lebih informatif saat memilih arsitektur model untuk penelitian serupa di masa mendatang.

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan tinjauan sistematis terhadap studi atau publikasi ilmiah yang relevan dengan topik penelitian ini. Penelitian terdahulu yang berkaitan disajikan pada Tabel 1.

Tabel 1 Penelitian Terdahulu

No.	Penelitian	Data	Metode	Hasil
1.	<i>K-Means and LSTM Combination for Social Media Black Campaign Identification of Indonesian Presidential Election 2024</i> (Priambodo dan Zuliarso, 2024)	Sumber: <i>Crawling</i> data Twitter (Tweet-Harvest). Jumlah: 8.620 data teks mentah, 7.025 data bersih setelah reduplikasi, dan 5.624 data setelah penyeimbangan kelas (2.812 per kelas). Label: <i>Black Campaign</i> dan <i>Non-Black Campaign</i> (dihasilkan melalui <i>clustering K-Means</i> dan pelabelan manual).	Preprocessing: Pembersihan teks ( <i>case folding</i> , <i>stemming</i> , dll.)  Pembobotan: TF-IDF  Pelabelan: <i>K-Means Clustering</i> Klasifikasi: <i>Long Short Term Memory (LSTM) Ensemble Learning</i> (sebagai pembandingan)  Evaluasi: Akurasi, Presisi, <i>Recall</i> , <i>F1-Score</i>	<i>Accuracy: 90.28%</i> , <i>Precision: 0.90</i> , <i>Recall: 0.91</i> , dan <i>F1-Score: 0.90</i>  <i>Model Ensemble (Pembandingan): Accuracy: 94.31%</i>

No.	Penelitian	Data	Metode	Hasil
2.	<i>Sentiment Classification on the 2024 Indonesian Presidential Candidate Dataset Using Deep Learning Approaches</i> (Suhaeni dkk., 2024)	Sumber: <i>Kaggle (Indonesian Presidential Candidate's Dataset, 2024)</i> . Jumlah: 6.757 data <i>tweet</i> (fokus pada Prabowo Subianto). Bahasa: Inggris Label: Sentimen Positif (5.013) dan Negatif (1.743)	<p><i>Preprocessing:</i> Pembersihan teks (<i>lowercasing</i>, menghapus <i>punctuation</i>, dll) .</p> <p><i>Splitting Data:</i> 60%, <i>Training</i>, 20% <i>Validation</i>, 20% <i>Testing</i></p> <p>Pembandingan: LSTM, <i>Bidirectional LSTM (BiLSTM)</i>, <i>Gated Recurrent Unit (GRU)</i></p> <p>Evaluasi: <i>Test Accuracy dan F1-Score</i></p>	<p><i>Model</i> BiLSTM: <i>Test Accuracy:</i> 80.70% dan <i>Test F1-Score:</i> 86.86%</p> <p><i>Model LSTM:</i> <i>Test Accuracy:</i> 72.56% dan <i>Test F1-Score:</i> 84.10%</p> <p><i>Model GRU:</i> <i>Test Accuracy:</i> 72.56% dan <i>Test F1-Score:</i> 84.10%</p>
3.	<i>Multilingual Transformer Contextual Embedding Model for Political Tweets Analysis</i> (Khatavkar et al., 2025)	Sumber: <i>Crawling Twitter (Tweepy)</i> dan dataset <i>Kaggle (Indian State Elections 2022)</i> . Jumlah: ±3 juta <i>tweet</i> (pelatihan) dan 500.000 <i>tweet</i> (eksperimen). Bahasa: Multilingual (Hindi, Inggris, bahasa regional India) dan menangani <i>code-switching</i> . Tugas: Klasifikasi sentimen, deteksi bias, dan deteksi emosi.	<p><i>Preprocessing:</i> <i>Tokenization</i> (BERT kustom), deteksi bahasa, identifikasi <i>code-switching</i>, normalisasi, dll.</p> <p><i>Modelling :</i> MTCE (<i>Multilingual Transformer Contextual Embedding</i>), yang merupakan ekstensi dari arsitektur BERT.</p> <p>Pembandingan: SVM, <i>Random Forest</i>, BERT.</p> <p>Evaluasi: <i>Accuracy, F1-Score, Precision, Recall, ROC-AUC.</i></p>	<p><i>Model MTCE:</i> <i>Accuracy:</i> 98.5%, <i>F1-Score:</i> 97.5% ,<i>Precision:</i> 98%, dan <i>Recall:</i> 97%</p> <p><i>Model BERT:</i> <i>Accuracy:</i> 97%</p> <p><i>Model</i> <i>Random Forest:</i> <i>Accuracy:</i> 90.5%</p>

No.	Penelitian	Data	Metode	Hasil
4.	<i>Indonesian multilabel classification using IndoBERT embedding and MBERT classification</i> (Nabiilah dkk., 2024)	Sumber: Komentar pengguna di berbagai media sosial Indonesia (Instagram, Twitter, dan Kaskus). Jumlah: 7.773 data. Label: Multi-label ( <i>Pornography, Hate Speech, Radicalism, Defamation</i> ).	<i>Preprocessing: Noise removal, case folding, translasi emoticon, tokenizing, translasi kata slang, stemming, dan stopword removal.</i>  <i>Feature Extraction: IndoBERT embedding</i>  <i>Klasifikasi: Multilingual BERT (MBERT).</i>  <i>Evaluasi: F1-Score, Akurasi.</i>	<i>Model IndoBERT: F1-Score: 0.9032</i>

Penelitian oleh Priambodo dan Zuliarso (2024) yang berjudul “*K-Means and LSTM Combination for Social Media Black Campaign Identification of Indonesian Presidential Election 2024*” bertujuan mengembangkan model untuk mendeteksi *Black Campaign* yang menargetkan calon presiden Indonesia 2024 di media sosial *Twitter*, karena hal tersebut dapat menyebabkan konflik dan merusak citra calon presiden. Metode yang digunakan adalah kombinasi algoritma *hybrid* yaitu *clustering K-Means* dan *Long Short-Term Memory (LSTM)*. *K-Means* diimplementasikan untuk mengelompokkan dataset yang sebelumnya melalui *preprocessing* dan pembobotan menggunakan *Term Frequency-Inverse Document Frequency (TF-IDF)* dan secara efektif mempersiapkan data untuk klasifikasi dengan menentukan label data menjadi *Black Campaign* dan *Non-Black Campaign*. Kemudian model LSTM yang merupakan jenis *Recurrent Neural Network (RNN)*, digunakan sebagai klasifikasi karena kemampuan LSTM mempelajari pola urutan kata dan memahami variasi kompleksitas dalam data teks sekuensial. Hasil klasifikasi menunjukkan model LSTM mencapai akurasi sebesar 90.28%. Dibandingkan dengan model *Ensemble Learning* sebagai pembanding, menghasilkan akurasi yang lebih tinggi yaitu 94,31%, penelitian ini menyimpulkan

bahwa LSTM mempunyai keunggulan yang lebih baik dalam memahami urutan kata dan konteks kalimat untuk deteksi *black campaign* dengan memanfaatkan kelebihan arsitektur *Deep Learning RNN*.

Penelitian oleh Suhaeni dkk. (2024) yang berjudul “*Sentiment Classification on the 2024 Indonesian Presidential Candidate Dataset Using Deep Learning Approaches*” bertujuan untuk membandingkan kinerja antara tiga model *deep learning*, yaitu (LSTM), *Bidirectional* (BiLSTM), dan *Gated Recurrent Unit* (GRU), dalam klasifikasi sentimen pada kumpulan data media sosial *X* yang terkait dengan calon presiden Prabowo Subianto pada Pemilihan Presiden Indonesia 2024. Model tersebut dievaluasi menggunakan metrik akurasi dan *F1-Score*. Hasil dari penelitian menunjukkan bahwa model BiLSTM secara konsisten unggul dari model LSTM dan GRU dalam semua metrik, yaitu mencapai akurasi pengujian tertinggi sebesar 80,70% dan *F1-Score* 86,86%. BiLSTM memiliki kinerja yang unggul dalam kemampuan untuk memahami konteks dua arah (*bidirectional context*) di dalam teks, yang memungkinkannya memahami pola sentimen kompleks secara lebih efektif. Kemudian model LSTM dan GRU menunjukkan kinerja yang identik dan lebih rendah, yaitu akurasi pengujian sebesar 72,56% dan *F1-Score* sebesar 84,10%. Oleh karena itu, penelitian ini menyimpulkan bahwa BiLSTM adalah model yang paling sesuai di antara ketiga model yang digunakan untuk analisis sentimen publik terhadap tokoh politik seperti Prabowo Subianto, sambil merekomendasikan eksplorasi model berbasis *transformer* seperti BERT atau GPT di masa mendatang untuk meningkatkan akurasi lebih lanjut.

Penelitian oleh Khatavkar *et al.* (2025) yang berjudul “*Multilingual Transformer Contextual Embedding Model for Political Tweets Analysis*” memperkenalkan model *Multilingual Transformer Contextual Embedding* (MTCE) yang inovatif untuk mengatasi tantangan analisis sentimen politik pada media sosial *X* di India, khususnya terkait dengan multilingualisme, *code-switching* (campuran bahasa Hindi, Inggris, dan dialek regional), serta deteksi emosi dan bias. Model MTCE yang dikembangkan berdasarkan arsitektur *transformer* seperti *BERT*, dirancang untuk memahami nuansa kontekstual dan kedalaman emosional dalam diskusi politik India. Model MTCE dilatih dan dievaluasi menggunakan data terdiri dari 3

juta *tweets* politik termasuk Hindi, Inggris dan bahasa daerah, atau sekitar 500.000 *tweets* yang dikumpulkan selama 6 bulan. Hasilnya menunjukkan bahwa model MTCE mencapai akurasi tertinggi sebesar 98,5% dan *F1-Score* sebesar 97,5%, melampaui model *state-of-the-art* yang ada, termasuk SVM, *Random Forest* dan model BERT dasar dengan rata-rata 2,5%. Keunggulan ini divalidasi secara statistik dan dikaitkan dengan kemampuan MTCE yang secara efektif dapat menangani kerumitan linguistik, *code-switching*, dan mendeteksi emosi yang kompleks seperti sarkasme (*F1-Score* 97%) yang menjadikannya model yang unggul untuk analisis sentimen politik yang mencakup di lanskap sosiopolitik India yang beragam bahasa.

Penelitian oleh Nabiilah dkk. (2024) bertujuan mengatasi maraknya *toxic* seperti rasisme, radikalisme dan pencemaran nama baik di media sosial Indonesia dengan menggunakan model klasifikasi *multilabel* yang baru. Keunikan penelitian ini terletak pada penggunaan dua model *pre-trained Transformer* yang berbeda untuk melakukan tugas yang berbeda seperti IndoBERT digunakan secara spesifik untuk tahap ekstraksi fitur (*embedding*), sementara MBERT (*Multilingual BERT*) digunakan untuk tahap klasifikasi akhir. Model ini diuji menggunakan 7.773 data komentar berbahasa Indonesia dari berbagai platform dan melalui proses *preprocessing* yang mendalam, termasuk translasi *emoticon* dan kata *slang*. Hasil akhir dari model gabungan IndoBERT dan MBERT yang digunakan ini terbukti paling optimal mencapai *F1-Score* sebesar 0.9032, serta menunjukkan performa yang lebih dan tidak rentan *overfitting* dibandingkan dengan penggunaan IndoBERT atau MBERT secara tunggal untuk kedua proses tersebut.

## **2.2 Black Campaign**

*Black campaign* merupakan bentuk strategi komunikasi yang bertujuan merusak reputasi suatu individu, organisasi, maupun brand melalui penyebaran informasi negatif, fitnah, atau berita palsu. Dengan berkembangnya teknologi dan penggunaan media sosial, praktik ini semakin mudah dilakukan. Hal tersebut sejalan dengan hasil survei Masyarakat Telematika Indonesia tahun 2022 yang menunjukkan bahwa media sosial menjadi saluran utama penyebaran hoaks.

Penelitian oleh Thanzani dkk. (2022) menyebutkan bahwa Kampanye Hitam (*Black Campaign*) merupakan tindakan dalam kegiatan kampanye yang dilarang dan berpotensi mengganggu keutuhan bangsa, karena bertujuan menjatuhkan lawan politik melalui penyebaran isu yang tidak berdasar atau tidak relevan. Karakteristik utamanya adalah penggunaan informasi yang lebih bersifat rekayasa daripada fakta, sering kali berupa fitnah tanpa bukti. Awalnya, praktik ini dikenal sebagai *whispering campaign* yang disebarkan secara lisan dari mulut ke mulut. Namun, seiring perkembangan teknologi dan globalisasi, bentuk kampanye ini telah bergeser ke ranah digital, terutama melalui media sosial seperti Twitter, Instagram, WhatsApp, dan TikTok. Di platform tersebut, praktik *Black Campaign* dapat berupa penghinaan, fitnah, perundungan, penyebaran hoaks, hingga penggunaan akun palsu untuk menyerang pihak tertentu.

*Black campaign* perlu dibedakan dari kampanye negatif (*negative campaign*). Kampanye Negatif masih diperbolehkan karena informasi yang disampaikan didasarkan pada fakta yang telah diverifikasi, meskipun bersifat menyerang pihak lawan. Sebaliknya, *black campaign* tidak hanya merugikan kandidat yang menjadi sasaran, tetapi juga masyarakat karena menyebarkan informasi palsu atau hoaks. Praktik ini tidak hanya mengancam nilai-nilai ideologi bangsa, tetapi juga memberikan dampak negatif terhadap kualitas pendidikan politik masyarakat Indonesia. Unsur-unsur yang dapat digolongkan sebagai *black campaign* berdasarkan ketentuan tersebut mencakup fitnah, pencemaran nama baik, dan penghinaan (Thanzani dkk., 2022).

Dalam konteks politik, *black campaign* atau kampanye hitam yaitu kampanye yang menyebarkan informasi tidak benar mengenai partai politik atau kandidat calon untuk menjatuhkan reputasi atau citra dan elektabilitasnya di mata publik. *Black campaign* memiliki ciri-ciri meliputi tuduhan tidak berdasarkan dengan fakta, mengandung unsur fitnah dan pencemaran nama baik, menggunakan media tidak resmi, dan bertujuan memengaruhi opini secara emosional untuk menjatuhkan. Indonesia juga memiliki aturan hukum yang berkaitan dengan *black campaign* yaitu Pasal 280 ayat (1) huruf c yang berbunyi mengenai larangan peserta pemilu menghina atau menjelekkan peserta pemilu yang lain dan Pasal 521 yang berisi

mengenai sanksi pidana penjara paling lama dua tahun dan denda paling banyak Rp24.000.000,00 (Komisi Pemilihan Umum Kabupaten Yalimo, 2025).

### ***2.3 Natural Language Processing***

Menurut penelitian oleh Wijaya dan Hayati (2025) *Natural Language Processing* atau Pemrosesan Bahasa Alami adalah cabang dari Kecerdasan Buatan (*Artificial Intelligence*) yang berfokus pada pemrosesan dan analisis bahasa alami manusia. Keunggulan utama NLP adalah kemampuannya memproses data skala besar dan mempelajari konteks bahasa yang rumit, termasuk sarkasme yang sulit ditangani oleh metode konvensional. Untuk mencapai akurasi tinggi, NLP memerlukan proses sistematis yang mencakup *preprocessing* data, *feature extraction* (seperti TF-IDF atau *Word Embedding*), serta penerapan model klasifikasi *Machine Learning* seperti *Naive Bayes* dan *Support Vector Machine* atau *Deep Learning* seperti LSTM atau BERT yang terbukti menawarkan efisiensi dan akurasi yang lebih baik daripada pendekatan tradisional.

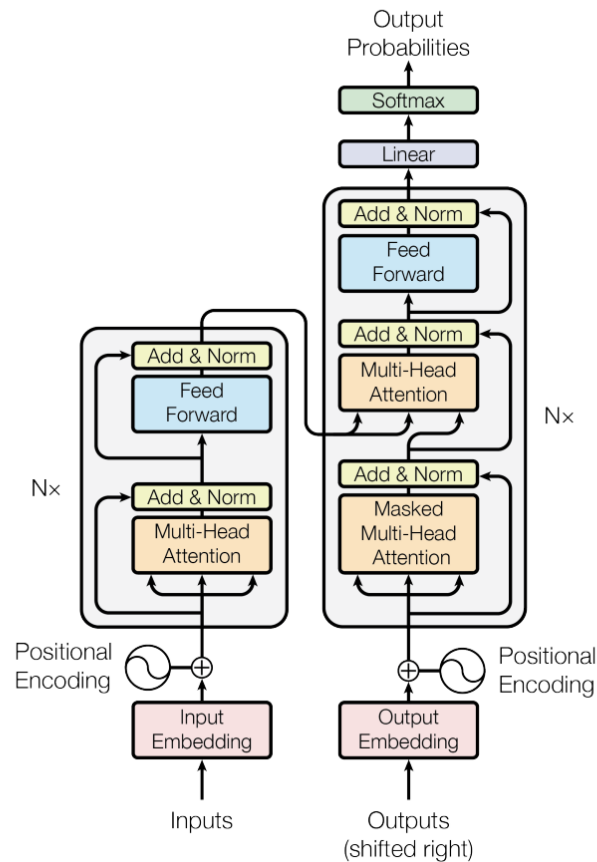
### ***2.4 Supervised Learning***

*Supervised Learning* atau pembelajaran terarah adalah salah satu jenis utama dalam *Machine Learning* di mana model dilatih menggunakan data yang sudah dilabel, sehingga *input* atau *ouput* yang diharapkan sudah diketahui sebelumnya. Dalam proses ini, sistem mempelajari hubungan antara *input* dan *output* berdasarkan contoh yang tersedia agar dapat melakukan prediksi terhadap data baru. Beberapa algoritma yang umum digunakan seperti *linear regression*, *logistic regression*, dan *neural network* (Pratama dkk., 2025).

### ***2.5 Transformer-based Model***

Model berbasis *Transformer* merupakan arsitektur model kecerdasan buatan (*artificial intelligence*) yang pertama kali diusulkan pada tahun 2017 kemudian diperkenalkan untuk menggantikan penggunaan model pendekatan sekuensial seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM)

yang mempunyai keterbatasan dalam pengolahan yang sulit dilakukan secara paralel. Model arsitektur *transformer* secara aktual menggantikan pendekatan sebelumnya di bidang Pengolahan Bahasa Alami (NLP), dengan mengambil mekanisme paralel dan *self-attention* atau atensi diri yang memungkinkan model untuk mengakses *input* secara langsung dan memperhitungkan ketergantungan antara kata dalam sekuens, yang sekaligus meningkatkan kecepatan pelatihan dan penalaran model. Mekanisme *self-attention* ini terintegrasi dalam lapisan *Multi-Head Attention* yang membantu model memahami representasi kata pada tingkat konseptual yang berbeda. Secara keseluruhan, model arsitektur *transformer* mampu meningkatkan efisiensi pelatihan, akurasi, serta pemahaman konteks panjang pada berbagai tugas NLP, yang kemudian membuka jalan bagi implementasi model yang lebih besar dan lebih kompleks seperti BERT dan GPT, yang menjadikannya model yang dominan dalam perkembangan dan aplikasi kecerdasan buatan atau *artificial intelligence* berbasis NLP (Firmanto dkk., 2024) .



Gambar 1 Arsitektur *Model Transformer* (Vaswani, 2017)

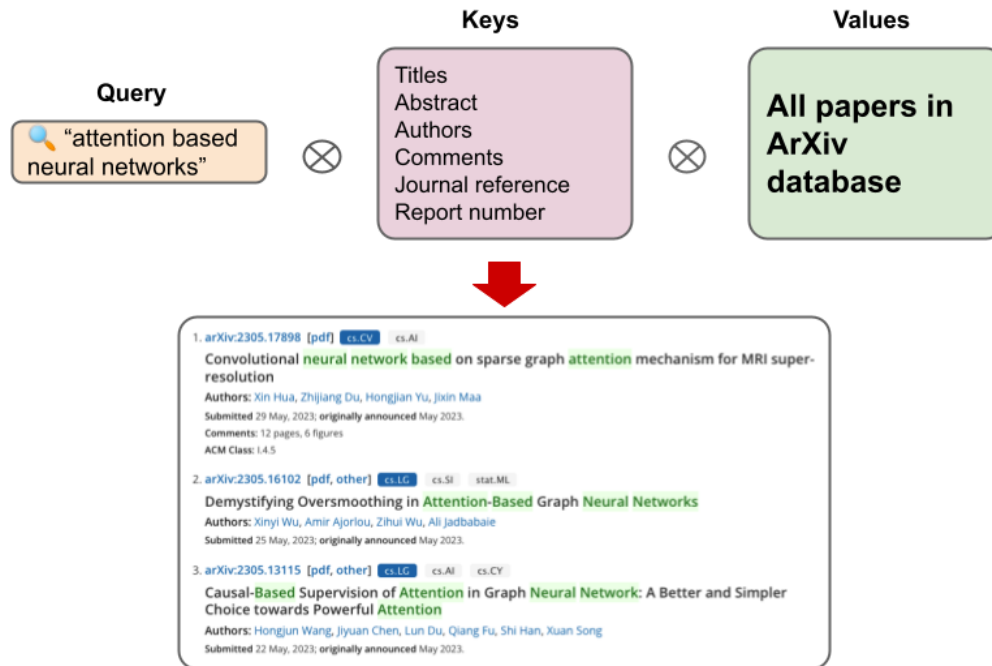
Menurut penelitian oleh Al-Faruq dan Aziz (2021), arsitektur *transformer* adalah model transduksi pertama yang dirancang untuk mengubah satu urutan lain dengan bantuan dua bagian utama yaitu *encoder* dan *decoder*. Model *transformer* ini sepenuhnya mengandalkan mekanisme *self-attention* untuk menghitung representasi input dan outputnya, sehingga tidak membutuhkan penggunaan RNN atau kompleksitas. Secara struktural, baik *encoder* ataupun *decoder* menggunakan lapisan *self-attention* dan lapisan *point-wise, fully connected* yang ditumpuk. Komponen yang penting dalam arsitektur ini yaitu *Multi-Head Attention* yang bertugas menghitung bobot *attention* dan terdiri dari 4 bagian yaitu lapisan linier, *scaled dot-product attention*, *concatenation*, dan lapisan linier akhir. Karena model *transformer* tidak secara alami memahami urutan, arsitektur ini juga menyertakan *Positional Encoding* yang di mana menggunakan fungsi sin dan cos dari frekuensi

yang berbeda untuk ditambahkan pada *embedding input*, yang berguna untuk memberikan informasi posisi setiap vektor kepada jaringan.

Dalam praktiknya, *Transformer* memproses beberapa urutan token sekaligus dalam satu *batch*. Agar setiap urutan memiliki panjang yang sama, dilakukan *padding* menggunakan token tambahan sehingga model dapat mengolahnya secara seragam dalam bentuk matriks. Token hasil *padding* ini nantinya diabaikan dalam mekanisme *self-attention*. Sebelum masuk ke lapisan perhatian, setiap *embedding* token juga ditambahkan informasi posisi melalui *positional embedding* karena *Transformer* tidak memiliki pemahaman posisi secara alami. Penambahan informasi ini membantu model mengenali urutan token dan memproses data dengan lebih akurat (Petru, 2023).

### 2.5.1 *Self-Attention*

Menurut penelitian oleh Chandradev dkk. (2022) menyebutkan bahwa mekanisme *self-attention* merupakan sebuah teknik mendasar pada model *Transformer* yang berfungsi untuk memodelkan hubungan kontekstual antar token dalam satu sekuens secara dinamis. Inti dari *self-attention* adalah kemampuannya dalam menilai relevansi setiap kata dalam kalimat terhadap kata lain yang sedang diproses, terlepas dari jarak di antara keduanya. Cara kerja mekanisme ini adalah dengan memproyeksikan representasi setiap token menjadi tiga vektor terpisah yaitu *Query* (Q), *Key* (K), dan *Value* (V). Skor perhatian (*attention score*) kemudian dihitung dengan mengukur kesamaan antara *Query* dari token saat ini dengan *Key* dari semua token lain (termasuk dirinya sendiri) dalam sekuens, Skor yang sudah dinormalisasi ini kemudian digunakan sebagai bobot untuk menjumlahkan semua vektor *Value* secara tertimbang (*weighted sum*). Hasilnya adalah sebuah representasi vektor baru untuk setiap token yang telah diperkaya dengan informasi kontekstual dari semua kalimat, memungkinkan model menangkap ketergantungan yang kompleks secara efektif.



Gambar 2 Ilustrasi *Query Key Value* (Nyandwi, 2023)

Secara intuitif, *attention* dapat dipahami sebagai mekanisme yang membuat model memusatkan perhatian pada bagian *input* yang paling penting. Secara teknis, *attention* bekerja dengan menghitung tingkat kesamaan antara dua vektor dan menghasilkan skor bobot berdasarkan kesamaan tersebut. Fungsi *attention* standar menggunakan tiga komponen utama, yaitu *Query*, *Key*, dan *Value*. Istilah *Query*, *Key*, dan *Value* sebenarnya dipinjam dari konsep pada mesin pencari atau sistem basis data. Untuk memberikan gambaran, bayangkan ketika seseorang mencari artikel bertema *attention* di *ArXiv*. *Query* adalah kata kunci yang dimasukkan ke kolom pencarian. Sistem kemudian mencocokkan *Query* tersebut dengan kumpulan *Key* yang telah diorganisasi di dalam basis datanya. Setelah mencocokkan kesamaan antara *Query* dan *Key*, sistem akan menampilkan *Value*, yaitu kumpulan artikel yang paling relevan dengan pencarian (Nyandwi, 2023).

### 2.5.2 Encoder Transformer

*Encoder* merupakan salah satu komponen utama dalam arsitektur *Transformer* yang berfungsi memproses urutan input pada tahap awal. Bagian ini bertugas mengubah rangkaian token mentah menjadi representasi yang lebih ringkas dan bermakna. Setiap blok *encoder* terdiri dari tiga komponen inti, yaitu *multi-head attention*

(MHA), *layer normalization*, dan *multilayer perceptron* (MLP) atau *feed-forward network*. MHA dan MLP berperan sebagai sub-lapis, yang dihubungkan menggunakan mekanisme *residual connection*, *dropout*, serta *layer normalization* untuk menjaga stabilitas proses pelatihan. Semakin banyak lapisan *encoder* yang digunakan, semakin besar pula kapasitas model untuk menangkap konteks global dalam sebuah urutan teks, sehingga berpotensi meningkatkan kemampuan generalisasi model terhadap berbagai jenis tugas (Nyandwi, 2023).

Menurut Petru (2023) komponen *encoder* pada arsitektur *Transformer* terdiri dari beberapa lapisan yang memiliki struktur seragam. Setiap lapisan memuat dua modul utama, yaitu *multi-headed self-attention* dan *feed-forward neural network*, yang masing-masing diikuti oleh proses *layer normalization* dan *residual connection* untuk menjaga stabilitas dan kualitas representasi. Ketika urutan input melewati lapisan-lapisan ini, setiap token akan diperbarui berdasarkan dua hal hubungan atau keterkaitannya dengan token lain yang relevan dalam urutan, serta transformasi non-linear yang dipelajari melalui jaringan saraf berlapis. Setelah melalui beberapa lapisan *encoder* yang identik, keluaran yang dihasilkan tetap memiliki panjang urutan yang sama namun dengan representasi token yang lebih kaya konteks. Peningkatan konteks ini merupakan hasil gabungan dari normalisasi, koneksi *residual*, dan mekanisme pemrosesan di setiap modul yang bekerja secara bertahap untuk membangun pemahaman yang lebih dalam terhadap *input*.

### **2.5.3 Decoder Transformer**

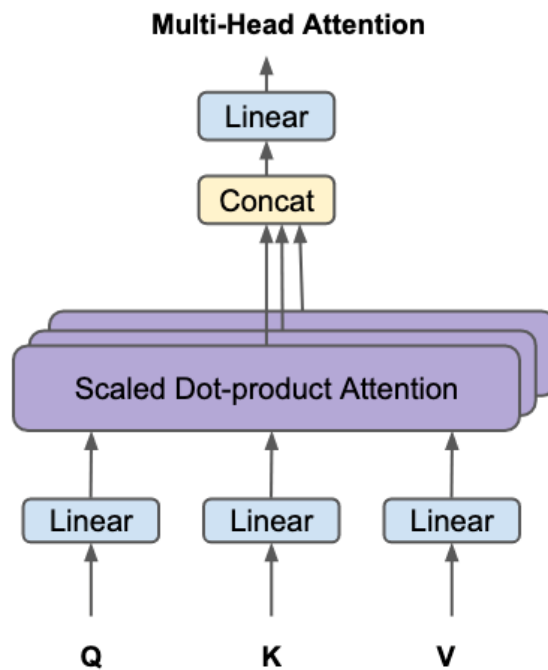
*Decoder* pada arsitektur *Transformer* memiliki struktur yang hampir sama dengan *encoder*, namun dilengkapi satu komponen tambahan berupa *multi-head attention* yang bekerja di atas keluaran *encoder*. Tujuan utama *decoder* adalah menggabungkan representasi dari *encoder* dengan urutan target untuk menghasilkan prediksi, seperti memutuskan token apa yang harus muncul berikutnya. Pada bagian *attention* yang memproses urutan target, diterapkan mekanisme *masked attention* untuk mencegah token yang sedang diproses melihat token setelahnya. Tanpa *masking*, model akan mengintip seluruh target dan menghasilkan prediksi yang tidak mencerminkan proses generatif yang sebenarnya,

sehingga berisiko gagal melakukan generalisasi. Sama seperti *encoder*, struktur *decoder* juga diulang beberapa kali (Nyandwi, 2023).

Menurut Petru (2023) *Decoder* pada *Transformer* terdiri dari beberapa lapisan identik yang masing-masing memuat *masked multi-head attention*, *multi-head attention*, dan *feed-forward neural network* (FFN). *Masked self-attention* memastikan setiap token hanya memperhatikan token sebelumnya sehingga model tidak mengintip ke depan saat melakukan prediksi. Selanjutnya, *multi-head attention* membantu menangkap beragam hubungan antar token, sementara FFN memberikan transformasi *non-linear* tambahan yang kemudian dinormalisasi. *Residual connection* juga disertakan untuk menjaga stabilitas informasi dan mencegah masalah seperti *vanishing gradient*. Melalui rangkaian proses ini, *decoder* menghasilkan representasi token yang kaya konteks dan siap digunakan untuk menghasilkan prediksi akhir.

#### **2.5.4 Multi-Head Attention**

*Multi-Head Attention* pada dasarnya merupakan pengembangan dari satu lapisan *attention*. Satu lapisan saja biasanya tidak cukup untuk menghasilkan representasi yang kuat, sehingga beberapa *attention* dihitung secara paralel lalu digabungkan. Setiap *head* di dalam *Multi-Head Attention* bekerja secara independen dengan melakukan perhitungan *attention* pada proyeksi linear dari vektor *Query*, *Key*, dan *Value* (QKV) (Nyandwi, 2023).



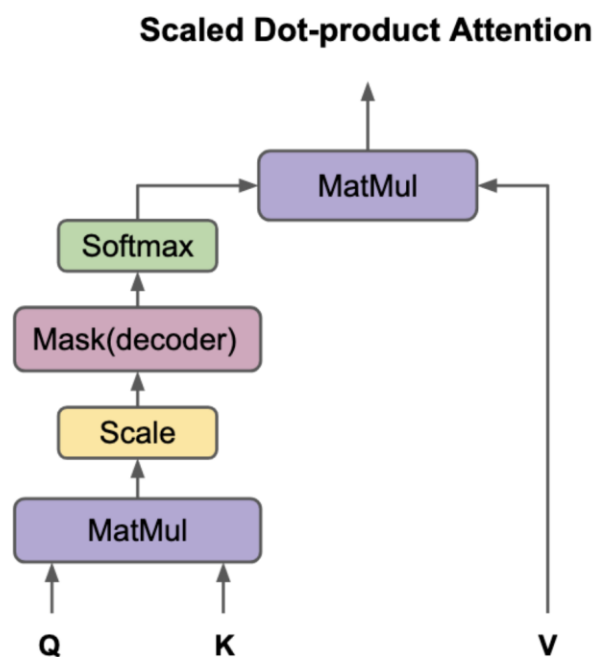
Gambar 3 Ilustrasi *Multi-Head Attention*

Hasil dari seluruh *head* kemudian digabungkan dan diproyeksikan kembali ke dimensi model. Pendekatan paralel ini memungkinkan model untuk menangkap informasi dari berbagai sudut pandang atau sub-ruang representasi secara bersamaan. Konsep *Multi-Head Attention* ini dapat dianalogikan dengan *depth-wise separable convolution* pada jaringan konvolusi, di mana *input* dipisah ke dalam beberapa *channel*, diproses secara mandiri, lalu digabung kembali melalui proyeksi linear. Pendekatan tersebut membuat proses lebih efisien sekaligus mampu memperkaya representasi fitur yang dihasilkan model (Nyandwi, 2023).

#### 2.5.4.1 *Scaled Dot-Product Attention*

Menurut penelitian oleh Prasetyo (2024) *Scaled Dot-Product Attention* merupakan komponen inti dari mekanisme *Self Attention* pada arsitektur *Transformer*. Mekanisme ini bekerja dengan memanfaatkan tiga vektor utama *Query*, *Key*, dan *Value* yang direpresentasikan dalam bentuk matriks  $Q$ ,  $K$ , dan  $V$ . Inti perhitungannya adalah menentukan skor perhatian antara *Query* dan *Key*, yang mencerminkan seberapa relevan suatu token terhadap token lainnya dalam urutan

*input*. Untuk menjaga stabilitas perhitungan *Softmax*, skor tersebut kemudian dibagi dengan akar kuadrat dari dimensi vektor *Query* sebagai langkah normalisasi (*scaling*). Skor yang telah dinormalisasi ini selanjutnya digunakan sebagai bobot untuk menghitung nilai rata-rata tertimbang dari matriks *value*, sehingga menghasilkan representasi baru yang memuat informasi konteks antar token. Ketika proses ini diterapkan pada seluruh pasangan *Query* dan *Key*, terbentuklah matriks perhatian yang menggambarkan hubungan antar-elemen dalam data, dan keluaran akhirnya adalah vektor representasi hasil *attention*.

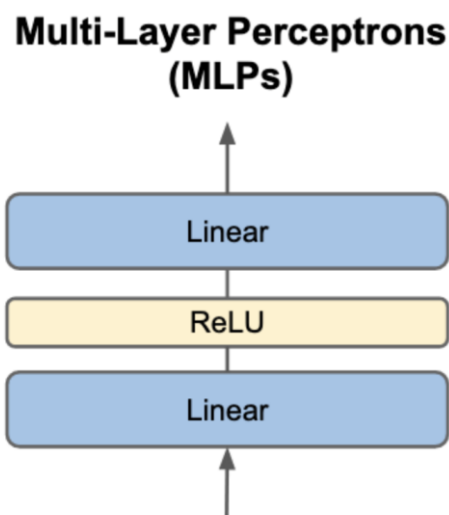


Gambar 4 Ilustrasi *Scaled Dot-product Attention* (Nyandwi, 2023)

### 2.5.5 Feed Forward

*Feed Forward Neural Network* (FFN) merupakan komponen penting dalam arsitektur *Transformer* yang diproses setelah *output* dari mekanisme *attention* diperoleh. Secara umum, FFN terdiri atas dua lapisan linear yang dipisahkan oleh fungsi aktivasi biasanya ReLU yang berfungsi menambahkan sifat non-linear sehingga model mampu mempelajari pola dan hubungan yang lebih kompleks dalam data. Pada setiap blok *Transformer*, FFN bekerja bersama *Residual Connection* dan *Layer Normalization*. *Residual connection* menjaga aliran gradien

tetap stabil sehingga proses pelatihan lebih efektif, sementara *Layer Normalization* membantu mempercepat konvergensi serta mempertahankan kestabilan distribusi nilai pada setiap lapisan. Kombinasi ketiga komponen ini memastikan bahwa representasi yang dihasilkan tetap informatif dan stabil selama proses pembelajaran (Darmawan dkk., 2025).



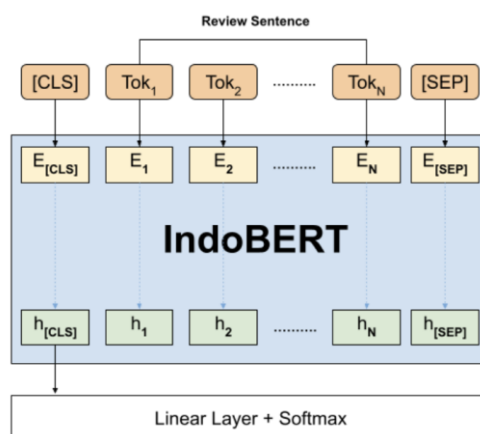
Gambar 5 Ilustrasi *Feed Forward* (Nyandwi, 2023)

## 2.6 Model berbasis *Transformer*

Untuk mengevaluasi arsitektur yang paling efektif dalam menangani karakteristik unik data penelitian ini, empat model berbasis *Transformer* dipilih untuk diuji, model tersebut antara lain:

### 2.6.1 *IndoBERT*

Penelitian oleh Ramadhan dkk. (2025) menyebutkan bahwa *IndoBERT* adalah model *pre-trained* berbasis arsitektur yang dikembangkan secara khusus untuk memproses Bahasa Indonesia. Keunggulan *IndoBERT* ada pada kemampuannya untuk memahami konteks kata dalam kalimat secara penuh dan menyeluruh, berkat mekanisme *self attention* dua arah dan pelatihannya yang spesifik pada Bahasa Indonesia.



Gambar 6 Arsitekur *IndoBERT* (Yulianti dan Nissa, 2024)

Proses pengolahan teks diawali dengan pemberian token khusus [CLS] pada awal kalimat yang berperan sebagai representasi keseluruhan input, serta token [SEP] yang digunakan sebagai penanda akhir atau pemisah antar kalimat. Teks yang telah ditokenisasi kemudian diproses melalui 12 lapisan pada model IndoBERT. Pada tahap akhir, vektor *hidden state* dari token [CLS] dipilih karena merepresentasikan informasi global dari seluruh urutan teks. Vektor ini selanjutnya diteruskan ke lapisan linear untuk memetakan fitur ke jumlah kelas yang ditentukan, dan hasilnya diolah menggunakan fungsi *softmax* untuk menghasilkan probabilitas pada setiap kelas sentimen. Selama proses *fine-tuning*, seluruh bobot model diperbarui agar sesuai dengan karakteristik data yang digunakan dalam penelitian (Yulianti dan Nissa, 2024).

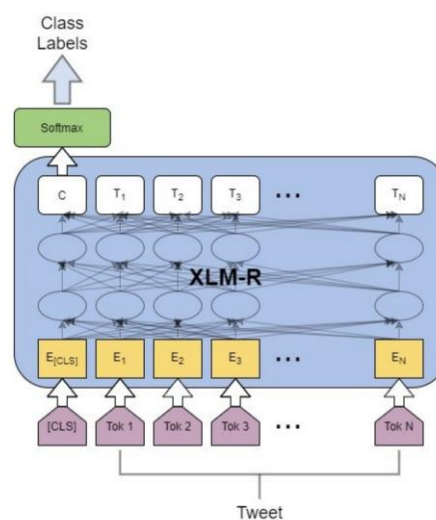
### 2.6.2 *IndoBERTtweet*

Menurut penelitian yang dilakukan oleh Damayanti dkk. (2025), Model ini hanya berbeda data latih dengan model *IndoBERT*. *IndoBERTtweet* merupakan model transformasi dari BERT (*Bidirectional Encoder Representations from Transformers*) yang sudah diadaptasi secara khusus untuk memproses teks Bahasa Indonesia di platform media sosial seperti Twitter, Model ini menggunakan pendekatan *domain-specific vocabulary adaptation*, di mana model ini menambahkan kosa kata khas Twitter seperti *slang*, *hashtag*, *emoticon*, dan *mention* ke dalam model *IndoBERT* yang sudah ada tanpa perlu melakukan *pre-training* dari

awal. Keunggulan model *IndoBERTtweet* terletak pada kemampuannya yang lebih efektif dalam mengenali, memproses, dan memahami struktur bahasa informal, singkatan, serta kalimat tidak baku yang ditemukan di media sosial Indonesia. Hal ini membuat model tersebut lebih unggul dalam menganalisis sentimen dibandingkan model BERT standar, sekaligus menyediakan peningkatan efisiensi hingga lima kali lebih cepat dibandingkan proses *pre-training* penuh.

### 2.6.3 XLM-RoBERTa

Penelitian oleh Agustini dkk. (2025) mengatakan bahwa *XLM-RoBERTa* (*Cross-Lingual Language Model*) merupakan model bahasa multibahasa (*multilingual*) berskala besar yang dikembangkan oleh Facebook AI. Model ini pada dasarnya adalah versi dari arsitektur *RoBERTa* yang dilatih pada dataset raksasa sebesar 2.5TB yang terdiri dari teks dalam 100 bahasa yang berbeda. Keunggulan utamanya adalah kemampuannya untuk memahami konteks lintas bahasa dengan sangat bagus, sehingga dapat mencapai kinerja canggih (*state-of-the-art*) dalam berbagai jenis tugas NLP seperti klasifikasi teks, *sequence labeling*, dan tanya jawab. Berbeda dengan model terdahulunya, *XLM-RoBERTa* terbukti sangat efektif bukan hanya untuk bahasa dengan sumber daya tinggi seperti Bahasa Inggris akan tetapi juga menunjukkan peningkatan performa yang signifikan untuk bahasa-bahasa dengan sumber daya terbatas (*low-resource languages*).



Gambar 7 Arsitektur *XLM-RoBERTa* (Ramanathan *et al.*, 2023)

Arsitektur *XLM-RoBERTa* merupakan pengembangan multibahasa dari model *Transformer RoBERTa* yang telah dilatih sebelumnya (*pre-trained*) pada 100 bahasa menggunakan pendekatan Masked Language Modeling (MLM). Model yang digunakan dalam penelitian ini memiliki sekitar 125 juta parameter dengan konfigurasi 12 lapisan, dimensi *hidden state* sebesar 768, *feed-forward hidden state* sebesar 3072, serta 8 *attention heads*. *XLM-RoBERTa* dirancang untuk memproses urutan input hingga 512 token, di mana token pertama selalu berupa token [CLS] yang berfungsi sebagai representasi khusus untuk keperluan klasifikasi dan menghasilkan representasi keseluruhan dari urutan teks (Ramanathan *et al.*, 2023).

## **2.7 Pre-processing**

Penelitian oleh Ardinata dkk. (2024) menyebutkan bahwa *pre-processing* merupakan tahapan yang dilakukan setelah data teks mentah berhasil dikumpulkan, karena data tersebut biasanya masih dalam format yang belum terstruktur. Data mentah biasanya masih terdapat banyak *noise* atau gangguan seperti emoji, *link*, dan berbagai karakter lain yang tidak relevan. Oleh karena itu, proses ini sangat diperlukan untuk membersihkan data mentah dengan tujuan akhir agar data yang dihasilkan hanya berupa teks saja dan siap untuk diolah.

### **2.7.1 Data Cleaning**

Dalam *pre-processing* teks, *data cleaning* merujuk pada tahapan spesifik untuk menghilangkan karakter-karakter pengganggu atau *noise*. Karakter yang umumnya dihapus meliputi angka, tautan, emoji, dan tanda baca. Proses pembersihan ini wajib dilakukan karena elemen-elemen tersebut dapat membuat data menjadi tidak efektif saat diproses. *Library regular expression* sering digunakan untuk membantu proses pembersihan ini (Ardinata dkk., 2024).

### **2.7.2 Case Folding**

*Case Folding* adalah proses untuk menyeragamkan setiap kata agar memiliki bentuk yang sama. Hal ini dilakukan untuk menghindari adanya perbedaan pada

kata-kata yang identik. Dalam praktiknya, seluruh huruf pada data teks akan diubah menjadi format huruf kecil (*lowercase*). Proses ini dapat diimplementasikan dengan mudah menggunakan *library Python* yang tersedia (Ardinata dkk., 2024).

### 2.7.3 Normalisasi Teks

Normalisasi *slang* adalah pendekatan untuk mengatasi penggunaan bahasa informal dalam teks sebelum diproses oleh NLP. Proses ini melibatkan penggantian kata *slang* dengan bentuk formalnya atau kaya lain yang lebih umum dipahami, sehingga teks yang dihasilkan menjadi lebih baku dan mudah diproses oleh model (Ardinata dkk., 2024).

### 2.7.4 Oversampling

Menurut Abidin dkk. (2025) *Oversampling* merupakan salah satu teknik penting yang digunakan untuk menangani permasalahan *imbalanced* dataset, yaitu kondisi ketika jumlah data pada kelas mayoritas jauh lebih besar dibandingkan kelas minoritas. Teknik ini dilakukan melalui proses augmentasi data dengan cara mengambil sampel dari kelas minoritas secara acak dengan pengembalian (*replacement*), kemudian menduplikasinya hingga jumlah data pada kedua kelas menjadi seimbang. Penerapan *oversampling* bertujuan untuk mengurangi kecenderungan model dalam memprediksi kelas mayoritas serta meningkatkan kemampuan model dalam mempelajari pola pada masing-masing kelas secara lebih optimal.

### 2.7.5 Downsampling

Menurut Murel (2024) *downsampling* merupakan metode yang digunakan untuk menyeimbangkan dataset dengan cara mengurangi jumlah sampel di kelas mayoritas agar setara dengan kelas minoritas. Teknik ini bermanfaat untuk mengurangi beban komputasi, mempercepat waktu pelatihan, dan membuang data yang berlebihan. Namun penelitian ini menyebutkan bahwa proses ini harus

dilakukan dengan hati-hati agar tidak menghilangkan informasi penting dan sampel yang dihasilkan tetap representatif terhadap keseluruhan data.

## **2.8 Splitting Data**

Menurut penelitian yang dilakukan oleh Pradana dkk. (2024), pemisahan data atau *splitting data* merupakan metode untuk membagi dataset yang sudah bersih menjadi dua bagian: data latih dan data uji. Tahapan ini krusial karena rasio pembagian yang tidak optimal berpotensi menurunkan akurasi model. Untuk menemukan perbandingan terbaik, penelitian ini melakukan eksperimen dengan lima skenario rasio yang berbeda, yakni 90:10, 80:20, 60:40, dan 50:50.

### **2.8.1 Data Train**

Data latih atau *data train* merupakan porsi data yang diperoleh dari hasil pemisahan data (*splitting*) yang dialokasikan untuk proses *modelling* atau pelatihan algoritma *Machine Learning* seperti *naive bayes* dan *neural network*. Ukuran data latih bervariasi tergantung rasio yang dipilih, contohnya dari total 365 data, rasio 80:20 memberikan 292 data latih, sementara rasio 50:50 memberikan 182 data latih. Dalam penelitian Pradana dkk. (2024) hasil evaluasi menunjukkan bahwa akurasi terbaik *naive bayes* (92%) tercapai saat menggunakan 80% data latih (rasio 80:20), sedangkan *neural network* mencapai akurasi puncaknya (91%) pada rasio 50% data latih (50:50).

### **2.8.2 Data Test**

Data uji (*data test*) adalah data yang dipisahkan bersama data latih, yang khusus digunakan untuk tahap evaluasi. Setelah model klasifikasi selesai dilatih menggunakan data latih, data uji akan diumpankan ke model tersebut menggunakan algoritma yang identik. Tujuan dari *testing* ini adalah untuk mengukur performa model dengan membandingkan prediksi yang dihasilkan dengan label data uji yang sebenarnya, biasanya melalui *confusion matrix* (Pradana dkk., 2024).

### 2.8.3 Data Validation

Menurut Gomstyn (2024), validasi data atau *data validation* adalah proses untuk memastikan bahwa data yang dimiliki sudah bersih, akurat, dan siap digunakan. Data dianggap valid jika memenuhi kriteria tertentu, seperti berada dalam format yang benar, masuk dalam rentang nilai yang diizinkan, bebas dari kesalahan, dan mematuhi aturan validasi spesifik yang ditetapkan oleh organisasi. Tujuan utama dari validasi ini adalah untuk menjamin kualitas data cukup tinggi untuk keperluan analisis dan pelatihan model AI, termasuk *machine learning*, sehingga data yang tidak valid tidak akan merusak hasil atau performa model.

## 2.9 Tokenisasi

Tokenisasi adalah sebuah teknik *preprocessing* esensial dalam NLP yang bertugas memecah data teks mentah menjadi unit-unit linguistik yang lebih kecil, seperti kata, klausa, atau kalimat. Proses ini sangat penting karena mengubah teks yang tidak terstruktur menjadi format yang dapat dipahami dan dianalisis secara efisien oleh algoritma komputer. *Token-token* yang dihasilkan dari proses ini merupakan representasi teks yang dapat diproses oleh algoritma, sehingga tokenisasi menjadi langkah dasar yang fundamental dalam *pipeline* pemrosesan bahasa alami (Holdsworth, 2024).

### 2.9.1 BertTokenizer

Menurut penelitian yang dilakukan oleh Lambruschini. (2023) menyebutkan *BertTokenizer* merupakan *tokenizer* spesifik yang digunakan oleh model *BERT* untuk melakukan proses tokenisasi data, yaitu mengubah teks mentah menjadi format numerik atau angka yang dapat dipelajari oleh model. Proses ini menggunakan pendekatan yang disebut *WordPiece*. Dalam metode *WordPiece*, *tokenizer* memanfaatkan kamus kata yang telah ditentukan, jika sebuah kata tidak ditemukan di dalam kamus, *tokenizer* akan memecah kata tersebut menjadi bagian yang lebih kecil yang dikenal sebagai *subword* atau *token-pieces*. Pendekatan ini memungkinkan model untuk secara fleksibel menangani kata-kata yang tidak umum atau baru. Kemudian *BertTokenizer* juga secara otomatis menambahkan

token-token khusus seperti [CLS] atau *classification* di awal input teks sebagai sinyal untuk klasifikasi, dan token [SEP] atau *separator* di akhir teks untuk menandai akhir tugas. Kemudian representasi dari token [CLS] inilah yang akan digunakan oleh lapisan *fully-connected* untuk menghasilkan prediksi label.

### 2.9.2 XLMRobertaTokenizer

Penelitian oleh Keenan Wiciaputra. (2021) menyebutkan bahwa *tokenizer* yang digunakan oleh XLM-R atau *XLMRoBERTaTokenizer* berbeda secara fundamental dari *tools* yang dipakai BERT dan XLM. XLM-R secara khusus melatih sebuah *Sentence Piece Model* (SPM) yang dapat diimplementasikan langsung ke data teks mentah dalam berbagai bahasa. Model ini dibuat menggunakan *library Sentence Piece* yang berfungsi sebagai *tokenizer* dan *detokenizer sub-word* independen. *Tokenizer* ini dirancang untuk pemrosesan teks berbasis *neural* dan dapat melakukan tokenisasi *sub-word* sekaligus mengonversi teks secara langsung menjadi urutan ide.

Menurut penelitian yang dilakukan oleh Salve *et al.* (2025), *Sentence Piece* merupakan *tokenizer sub-word* yang bersifat *language unaware* atau tidak bergantung pada bahasa, sehingga ideal untuk pemrosesan teks berbasis *neural*, terutama dalam konteks *multilingual* (banyak bahasa) dan *code-mixed* (campur kode). Keunggulannya terletak pada kemampuannya untuk bekerja langsung pada teks mentah (*raw text*), tidak seperti *tokenizer* tradisional yang memerlukan pra-tokenisasi berdasarkan spasi. Fitur ini membuatnya sangat bermanfaat untuk bahasa tanpa batas kata yang jelas seperti Jepang atau Mandarin dan teks *code-switched* seperti *Taglish* yang aturan spasinya tidak konsisten.

### 2.10 Fine-Tuning

*Fine-tuning* merupakan salah satu pendekatan utama dalam pemanfaatan model *Transformer* setelah tahap *pre-training*, dan dipilih dalam penelitian ini sebagai strategi untuk menyelesaikan masalah deteksi ulasan palsu. Secara umum, *fine-tuning* adalah proses melatih ulang model yang telah melalui *pre-training* dengan

menginisialisasi seluruh parameternya, lalu melanjutkan pelatihannya menggunakan data berlabel. Pendekatan ini semakin populer karena terbukti lebih efisien dari segi waktu, biaya, dan kebutuhan komputasi dibandingkan melatih model dari awal (Awalina dkk., 2022).

### 2.10.1 *Hyperparameter Tuning*

Penelitian oleh Kania dkk. (2025) menyebutkan bahwa *Hyperparameter tuning* merupakan langkah penting dalam proses *machine learning* karena menentukan bagaimana sebuah model bekerja sebelum mulai belajar dari data. Tidak seperti parameter internal yang otomatis dipelajari selama pelatihan, *hyperparameter* harus ditentukan sejak awal dan sangat memengaruhi kemampuan model dalam mengenali pola tanpa terlalu menghafal data atau justru gagal memahami struktur yang ada. Pemilihan nilai yang kurang tepat dapat membuat model terlalu kompleks sehingga *overfitting*, atau terlalu sederhana sehingga *underfitting* dua kondisi yang sama-sama menurunkan akurasi saat model diuji pada data baru. Dalam konteks penelitian ini, penyesuaian *hyperparameter* terbukti membantu meningkatkan performa model klasifikasi dan menghasilkan prediksi risiko penyakit jantung yang lebih akurat dan dapat diandalkan.

### 2.11 *Hyperparameter*

Menurut penelitian oleh Ilemobayo *et al.* (2024), *Hyperparameter* merupakan pengaturan yang mengontrol proses pelatihan serta struktur dari sebuah model *machine learning*. Pengaturan ini harus ditentukan terlebih dahulu sebelum pelatihan dimulai, berbeda dengan parameter model yang nilainya justru dipelajari selama proses pelatihan. Penetapan *hyperparameter* memiliki peran krusial dalam menentukan performa akhir dari model tersebut. Beberapa contohnya meliputi *learning rate* (laju pembelajaran) pada *neural network*, jumlah pohon (*trees*) pada *random forest*, kedalaman *decision tree*, nilai penalti pada *support vector machine*, momentum, *learning rate decay* (peluruhan laju pembelajaran), dan konstanta regularisasi.

### 2.11.1 *Epoch*

Menurut Johannes dkk. (2024) *Epochs* adalah satu siklus hitungan yang selesai setelah seluruh data sampel (semua sampel) telah digunakan dalam proses pelatihan setidaknya satu kali. Dalam praktiknya, data dibagi menjadi kelompok-kelompok kecil yang disebut *batch size* (ukuran set sampel yang dipakai pada satu kali *training*). Sebagai ilustrasi, jika 80 sampel dibagi menjadi 5 *batch* (dengan *batch size* 16), maka 1 *epoch* dianggap selesai setelah kelima *batch* tersebut selesai dilatih.

### 2.11.2 *Batch Size*

Penelitian oleh Ilemobayo *et al.* (2024) menyebutkan bahwa *Batch size* merujuk pada jumlah data sampel yang dipakai untuk menghitung gradien dalam satu langkah (iterasi) pelatihan, khususnya dalam metode *stochastic gradient descent* (*SGD*). Ada konsekuensi dalam pemilihannya yaitu jika ukurannya kecil, estimasi gradien yang didapat bisa jadi lebih presisi, namun kelemahannya adalah proses pelatihan (konvergensi) mungkin membutuhkan waktu lebih lama karena memerlukan lebih banyak iterasi.

### 2.11.3 *Learning Rate*

*Learning-rate* adalah *hyperparameter* esensial dalam *deep learning* yang berfungsi sebagai pengatur ukuran langkah (*step size*) dalam proses optimisasi. Secara sederhana, *hyperparameter* ini mengontrol seberapa besar penyesuaian yang dilakukan pada bobot model berdasarkan kesalahan yang terdeteksi saat pelatihan. Menentukan nilainya sangat penting, karena jika terlalu besar, model bisa melompat melewati solusi terbaik (konvergensi sub-optimal), dan jika terlalu kecil, proses pelatihan akan memakan waktu yang sangat lama (Ilemobayo *et al.*, 2024).

### 2.11.4 *Early Stopping*

*Early Stopping* merupakan teknik yang digunakan untuk mencegah *overfitting* sekaligus menentukan titik optimal penghentian pelatihan model. Metode ini

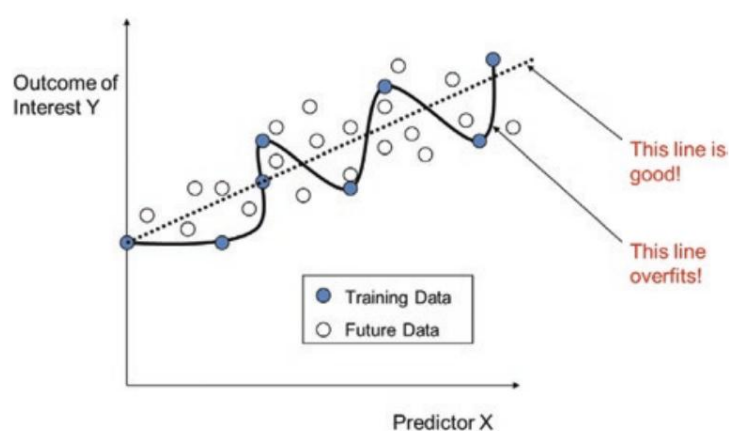
diterapkan sebagai langkah pengurangan risiko tambahan bersama *Dropout* guna menghindari kondisi ketika model memiliki akurasi tinggi pada data pelatihan tetapi berkinerja buruk pada data yang belum pernah dilihat. *Early Stopping* membantu menjaga kemampuan generalisasi model dengan menghentikan pelatihan sebelum model terlalu menyesuaikan diri terhadap data pelatihan. Secara operasional, teknik ini bekerja dengan memonitor *validation loss* dan menghentikan pelatihan ketika tidak terjadi peningkatan selama lima *epoch* berturut-turut. Dengan demikian, model dapat berhenti pada titik yang paling efektif, sehingga tercapai keseimbangan antara *underfitting* dan *overfitting* (Anam dkk., 2024).

### 2.11.5 *Weight Decay*

Menurut Pan (2020) *Weight Decay (WD)* merupakan teknik regulasi penting dalam pelatihan model *deep learning* yang berfungsi mengendalikan besarnya bobot pada jaringan saraf. Secara teknis, WD diterapkan dengan menambahkan istilah penalti pada fungsi *loss* akhir, yang dihitung berdasarkan norma *Frobenius* dari setiap matriks bobot serta vektor bobot pada *Batch Normalization (BN)*. Secara konvensional, WD berperan sebagai pembatas kapasitas jaringan dan membantu meningkatkan kemampuan generalisasi model dengan menekan komponen bobot yang tidak diperlukan. Namun, penelitian ini memberikan perspektif baru dengan menunjukkan bahwa WD dan BN memiliki peran penting dalam membentuk struktur geometris *Neural Collapse (NC)*, yaitu kondisi ketika representasi fitur pada lapisan akhir untuk kelas yang sama berkumpul pada satu titik. Dalam konteks ini, WD berfungsi sebagai katalis yang mendorong terbentuknya fitur yang lebih kompak untuk kelas yang sama. Temuan teoretis dan empiris menunjukkan bahwa penggunaan nilai WD yang lebih tinggi (hingga batas tertentu) memberikan jaminan konsistensi yang lebih kuat terhadap terjadinya NC, karena WD mengatur norma fitur serta matriks bobot, dan tingkat NC paling jelas muncul pada model yang menggunakan BN dengan nilai WD yang besar.

### 2.12 *Overfitting*

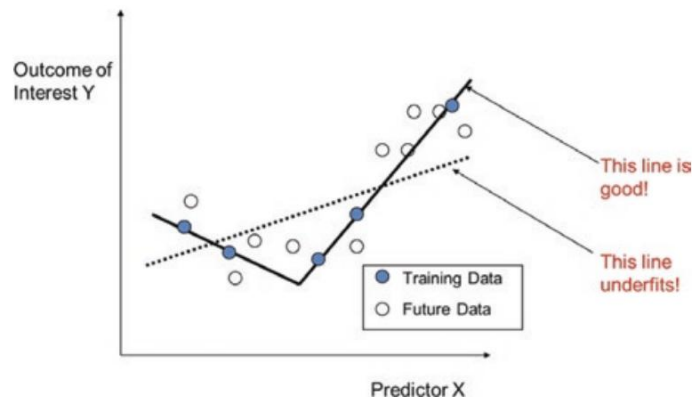
*Overfitting* terjadi ketika sebuah model sangat cocok dengan data pelatihannya, namun kinerjanya buruk saat dihadapkan pada data baru dari distribusi yang sama. Hal ini terjadi karena model tersebut tidak hanya mempelajari pola yang relevan, tetapi juga *noise* atau keunikan yang hanya ada di data pelatihan dan tidak mewakili populasi data yang sesungguhnya. Model yang *overfitting* sering juga didefinisikan sebagai model yang memiliki kompleksitas lebih tinggi daripada yang idealnya diperlukan untuk masalah tersebut (Aliferis and Simon, 2024).



Gambar 8 Ilustrasi *Overfitting* (Aliferis and Simon, 2024)

### 2.13 *Underfitting*

*Underfitting* adalah situasi di mana model gagal menangkap pola yang mendasari data, sehingga kinerjanya buruk tidak hanya pada data populasi umum tetapi juga pada data pelatihan itu sendiri (sub-optimal). Model yang *underfitting* biasanya terlalu sederhana. Secara lebih luas, model ini memiliki kesalahan generalisasi yang lebih besar daripada kesalahan yang dimiliki oleh model terbaik yang mungkin bisa dibuat dari data tersebut. Akibatnya, kemampuan prediktifnya akan rendah baik pada data latih maupun data di masa depan (Aliferis and Simon, 2024).



Gambar 9 Ilustrasi *Underfitting* (Aliferis and Simon, 2024)

### 2.14 Confusion Matrix

*Confusion Matrix* adalah sebuah matriks dua dimensi yang membandingkan label sebenarnya dengan label yang diprediksi oleh *classifier*. Fungsi utamanya adalah untuk menyajikan distribusi hasil prediksi ke semua kelas dalam satu tampilan yang ringkas. Jika matriks ini dinormalisasi berdasarkan barisnya, maka persentase *False Negative* (FN) untuk setiap kelas dapat terlihat. Selain itu, sel-sel yang terletak di diagonal utama pada matriks yang telah dinormalisasi tersebut menunjukkan nilai *recall* untuk masing-masing kelas Heydarian and Doyle (2022). Selain itu, *confusion matrix* memungkinkan perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengukur kualitas prediksi yang dihasilkan oleh model klasifikasi. Tabel *confusion matrix* dapat dilihat pada Tabel 2.

Tabel 2 *Confusion Matrix* (Maxwell and Warner, 2021)

		<i>Predicted Class</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Actual Class</i>	<i>Negative</i>	TP	FN
	<i>Positif</i>	FP	TN

Keterangan:

1. *True Positive* (TP) = Data positif yang terdeteksi positif
2. *False Positive* (FP) = Data negatif yang terdeteksi positif
3. *True Negative* (TN) = Data positif yang terdeteksi negatif

4. *False Negative* (FN) = Data negatif yang terdeteksi positif

*Confusion matrix* biner memiliki empat entri yaitu jumlah sampel *true positive* (TP) dan *true negative* (TN), yang masing-masing adalah sampel yang dipetakan dengan benar sebagai positif dan negatif, dan dua kategori kesalahan yaitu sampel *false positive* (FP) dan *false negative* (FN), yang mewakili jumlah negatif yang salah dipetakan sebagai positif, dan sebaliknya. Dalam pengujian hipotesis statistik, FP disebut sebagai *error* Tipe I, dan FN sebagai *error* Tipe II (Maxwell and Warner, 2021).

#### 2.14.1 Accuracy

Nilai akurasi membantu untuk menilai performa prediksi total sebuah model klasifikasi. Akan tetapi, penting untuk diingat bahwa akurasi saja tidak bisa diandalkan sebagai evaluasi jika dataset yang digunakan tidak seimbang Paraijun dkk. (2022). Rumus yang digunakan untuk mengukur tingkat *Accuracy* dapat dilihat pada rumus (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (1)$$

#### 2.14.2 Precision

*Precision* adalah kemampuan model *classifier* untuk menghindari kesalahan saat menebak sesuatu sebagai positif. Sederhananya, ini mengukur seberapa akurat tebakan positif dari model. Tujuannya adalah untuk tidak salah melabeli sesuatu yang sebenarnya negatif sebagai positif Paraijun dkk. (2022). Rumus yang dipakai untuk menghitung *Precision* dapat dilihat pada rumus (2).

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (2)$$

#### 2.14.3 Recall

*Recall* adalah kemampuan model (*classifier*) untuk menemukan semua sampel yang memang *seharusnya* positif. Ini mengukur seberapa tuntas model dalam

mengidentifikasi kelas positif dari semua yang seharusnya positif di dalam data, berapa banyak yang berhasil ditemukan oleh model Paraijun dkk. (2022). Rumus yang digunakan untuk menghitung *Recall* dapat dilihat pada rumus (3).

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (3)$$

#### 2.14.4 *F1-Score*

*F1-Score* adalah sebuah metrik yang menyeimbangkan *precision* dan *recall* dengan cara menggabungkan keduanya ke dalam satu nilai. Metrik ini sangat bermanfaat untuk menilai performa model pada dataset yang tidak seimbang (*imbalanced*), karena *F1-Score* memperhitungkan *false positive* sekaligus *false negative* Paraijun dkk. (2022). Rumus yang digunakan untuk menghitung *F1-Score* dapat dilihat pada rumus (4).

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots\dots\dots (4)$$

## **BAB III METODOLOGI PENELITIAN**

### **3.1 Tempat dan Waktu**

#### **3.1.1 Tempat**

Penelitian ini dilaksanakan di lingkungan Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, Jalan Prof. Dr. Ir. Sumantri Bojonegoro No.1, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Provinsi Lampung.

#### **3.1.2 Waktu**

Penelitian ini dilaksanakan pada semester ganjil tahun akademik 2025/2026, dimulai dari bulan Oktober 2025 hingga bulan Januari 2026.

### **3.2 Perangkat Penelitian**

Untuk menunjang pelaksanaan penelitian ini, dibutuhkan perangkat keras dan perangkat lunak sebagai berikut:

#### **3.2.1 Perangkat Keras**

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut:

1. *System Manufacture* : ASUS
2. *System Model* : ASUS Vivobook K3402ZA
3. *Processor* : Intel® Core™ i5-12500H
4. *Graphic* : Intel Iris Xe Graphics
5. *Memory* : 12 GB RAM
6. *Storage* : SSD 512 GB

### 3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Sistem Operasi Windows 11 Home 64-bit
2. Google Colab
3. Google Drive
4. GitHub
5. Kaggle

### 3.3 Dataset

Dataset yang digunakan pada penelitian ini bersumber dari riset yang dilakukan oleh Priambodo dan Zuliarso (2024) yang dipublikasikan dalam artikel yang berjudul “*K-Means and LSTM Combination for Social Media Black Campaign Identification of Indonesian Presidential Election 2024*”. Dataset tersebut sudah dilakukan tahap *clustering* menggunakan *K-Means* sebanyak 7025 baris, dan memiliki label dengan dua kelas yaitu *Black Campaign* sebanyak 4213 baris dan *Non-Black Campaign* sebanyak 2812 baris. Contoh dataset penelitian dapat dilihat pada Tabel 3 berikut:

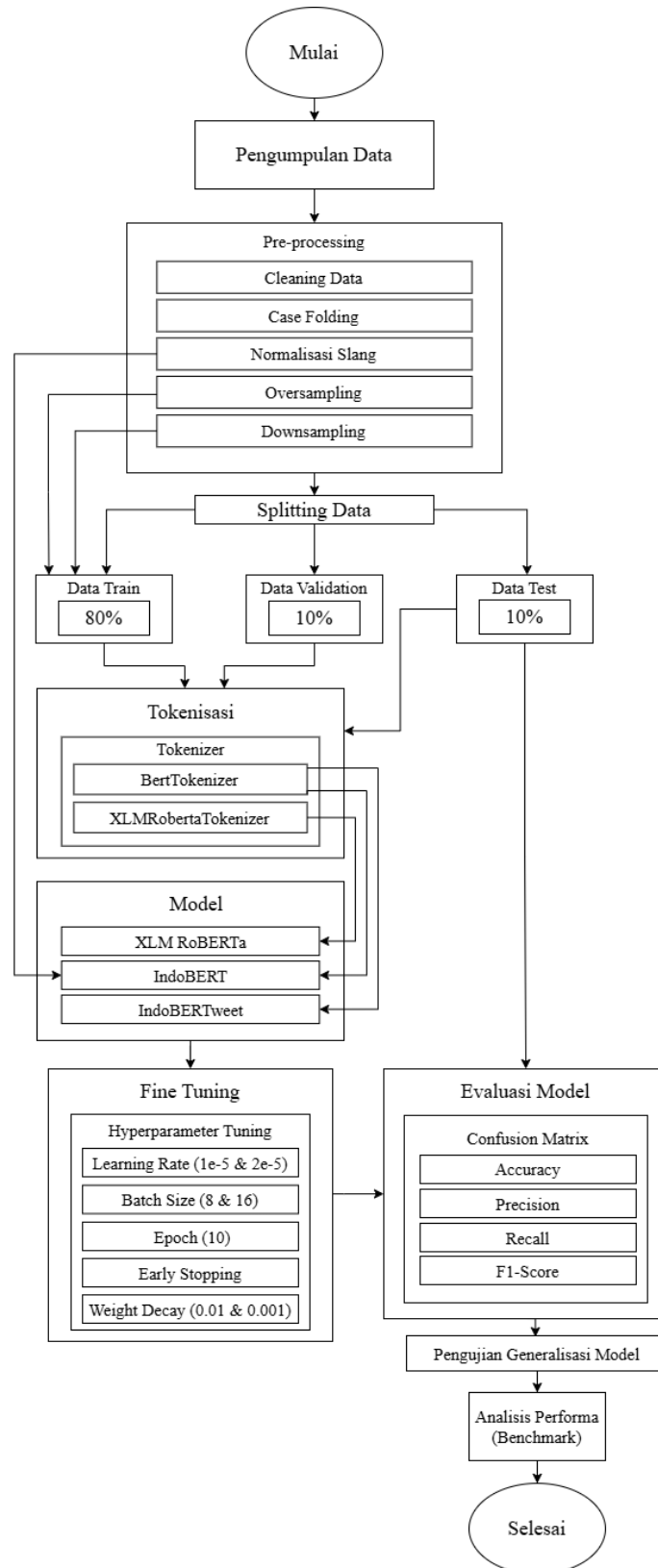
Tabel 3 Contoh Data Penelitian

<i>No.</i>	<i>Date</i>	<i>Username</i>	<i>Text</i>	<i>Label</i>
1.	Thu Oct 19 22:07:12	Tohid_siswanto	@OposisiOjol @AdeSudr90969850 @ganjarpranowo @mohmahfudmd Calon Presiden rakyat ya yg anter rakyat.. Bukan seperti wan	<i>black-</i> <i>campaign</i>

No.	Date	Username	Text	Label
	+0000 2023		Toa..masa yg hadir kader partai dan hulu balangnya.. NGIBULnya ga ketulungan.. Ngurus jakarta aja ga becus Eh pakai mau selamatin palestine <a href="https://t.co/QcOLFulhcJ">https://t.co/QcOLFulhcJ</a>	
	Thu Oct 19		#AMINKanIndonesia #AniesMuhaimin2024 Siap	
2.	09:19:41 +0000 2023	hari1234t678910	menyongsong perubahan untuk Indonesia lebih baik <a href="https://t.co/iWHMvN3Cmo">https://t.co/iWHMvN3Cmo</a> @ch_chotimah2 @ganjarpranowo @mohmahfudmd Halah siapa yg masih percaya sama pdi kecuali orang patah hati „korupsi juara„nangkep harun masiku ga becus..pak mpud semangat uang ratusan trilyun sekarang lupa to ampun ampun jangan sampai dagelan di negri ini ada lagi	<i>nonblack-campaign</i>
	Thu Oct 19		@voiceofpkb @aniesbaswedan @cakimiNOW Tetap semangat gus	
3.	08:47:27 +0000 2023	obor2150	@cakimiNOW @aniesbaswedan #AMINKanIndonesia #aniesmuhaimin2024	<i>black-campaign</i>
	Thu Oct 19		@HusinShihab @mohmahfudmd @ganjarpranowo Ngurus duit 349T	
4	06:04:52 +0000 2023	JokoGadhang	#AMINKanIndonesia #aniesmuhaimin2024	<i>nonblack-campaign</i>
	Thu Oct 19		@HusinShihab @mohmahfudmd @ganjarpranowo Ngurus duit 349T	
5	00:18:09 +0000 2023	mustafaagus007	aja kagak becus... cuma gede bacot di dpn anggota DPR <a href="https://t.co/KQz5OILL9Y">https://t.co/KQz5OILL9Y</a>	<i>black-campaign</i>

### 3.4 Tahapan Penelitian

Alur dari tahapan yang akan dilakukan pada penelitian ini diilustrasikan pada Gambar 8.



Gambar 10 Tahapan Penelitian

Penelitian dilakukan dalam tujuh tahap, yaitu Pengumpulan Data, *Pre-processing*, *Splitting Data*, Tokenisasi, *Fine-Tuning*, Evaluasi Model, dan Analisis Performa (*Benchmark*).

### 3.4.1 Pengumpulan Data

Pengumpulan data dalam penelitian ini berasal dari sumber sekunder yaitu data penelitian oleh Priambodo dan Zuliarso (2024) yang diperoleh melalui repositori GitHub, data tersebut digunakan sebagai dasar tahap pada penelitian ini.

### 3.4.2 *Pre-processing*

Tahap *preprocessing*, ini sangat diperlukan untuk membersihkan data mentah dengan tujuan akhir agar data yang dihasilkan hanya berupa teks saja dan siap untuk diolah. Berikut tahapan *preprocessing* yang dilakukan pada penelitian ini:

1. *Cleaning Data* : Tahap ini bertujuan untuk membersihkan data dari *noise* seperti *hashtag*, URL, dan simbol.
2. *Case Folding* : Tahap ini bertujuan menyeragamkan teks dengan mengubah seluruh karakter menjadi huruf kecil, sehingga kata yang sama tidak dianggap berbeda hanya karena variasi kapitalisasi.
3. Normalisasi *Slang* : Tahap ini menyeragamkan kata-kata slang atau informal menjadi bentuk baku menggunakan kamus *alay*. Langkah ini penting agar model dapat memahami makna teks secara konsisten, karena variasi kata informal sering tidak dikenali oleh model yang dilatih pada data formal. Normalisasi hanya dilakukan pada model *IndoBERT* yang dilatih pada korpus bahasa formal. normalisasi tidak diterapkan pada *IndoBERTtweet* dan *XLNet-RoBERTa* karena *IndoBERTtweet* secara khusus dilatih menggunakan data Twitter yang didominasi oleh bahasa informal, sedangkan *XLNet-RoBERTa* didukung oleh *tokenizer* berbasis *SentencePiece* serta korpus *CommonCrawl* berskala besar, yang membuatnya memiliki tingkat ketahanan tinggi terhadap variasi bahasa dan *noise* tanpa memerlukan proses penyeragaman tambahan yang berpotensi menghilangkan karakteristik asli data media sosial.

4. *Oversampling* : Tahap ini dilakukan untuk mengatasi ketidakseimbangan kelas pada dataset latih. Pada data mentah, kelas *Black Campaign* memiliki 4.213 sampel, sedangkan *Non-Black Campaign* berjumlah 2.812 sampel. Untuk menyeimbangkannya, jumlah sampel pada kelas minoritas pada data *train* ditambah secara acak hingga menyamai kelas mayoritas. *Oversampling* juga membantu mengurangi risiko *overfitting* terhadap pola yang dominan pada kelas mayoritas dan membuat proses pembelajaran lebih fokus pada fitur pembeda antar kelas. Dengan distribusi yang lebih seimbang, model dapat mempelajari perbedaan kedua kelas secara lebih efektif dan menghasilkan prediksi yang lebih adil serta stabil.
5. *Downsampling*: Tahap ini dilakukan untuk mengatasi ketidakseimbangan kelas pada dataset. Pada data mentah, kelas *Black Campaign* memiliki 4.213 sampel, sedangkan *Non-Black Campaign* berjumlah 2.812 sampel. Untuk menyeimbangkannya, jumlah sampel pada kelas mayoritas pada data *train* dikurangi secara acak hingga menyamai kelas minoritas. Sehingga model dapat belajar secara lebih adil dari kedua kelas. *Downsampling* juga membantu mengurangi risiko *overfitting*.

### 3.4.3 Splitting Data

Pembagian dataset merupakan langkah fundamental dalam *Machine Learning* untuk menilai kinerja model secara objektif sekaligus mencegah terjadinya *overfitting*. Pada penelitian ini, dataset hasil *preprocessing* yang telah seimbang kemudian dibagi ke dalam tiga subset, yaitu data latih, data validasi, dan data uji, dengan menerapkan *stratified sampling* agar proporsi kelas tetap konsisten, yakni 50% *Black Campaign* dan 50% *Non-Black Campaign*, kemudian dibagi menggunakan rasio 80% untuk data latih, serta masing-masing 10% untuk data validasi dan data uji. Pemilihan proporsi ini mempertimbangkan karakteristik model Transformer yang digunakan, seperti *IndoBERT*, *IndoBERTweet*, dan *XLNet*, yang memiliki jumlah parameter besar sehingga membutuhkan data latih yang cukup untuk mendukung proses *fine-tuning*, sementara porsi data validasi dan data uji dinilai memadai untuk menghasilkan evaluasi performa model yang representatif.

### 3.4.4 Tokenisasi

Tokenisasi merupakan tahap dasar dalam NLP yang berfungsi mengubah teks mentah menjadi representasi numerik yang dapat diproses model *Transformer* dengan cara memecah kalimat menjadi token (kata atau sub-kata). Penelitian ini menggunakan fungsi *AutoTokenizer* dari pustaka *Hugging Face* untuk memuat *tokenizer* yang sesuai dengan arsitektur masing-masing model. Untuk model *indobenchmark/IndoBERT-base-pl* dan *indolem/IndoBERTweet-base-uncased*, sistem otomatis memanggil *BERTTokenizer*, sedangkan *XLMRoBERTaTokenizer* digunakan untuk *xlm-roBERTa-base*. Pemilihan *BERTTokenizer* yang berbasis algoritma *WordPiece* dianggap tepat karena mampu menangani masalah *Out-of-Vocabulary* (OOV) pada teks *black campaign* yang cenderung informal. *WordPiece* memecah kata tidak baku menjadi *subwords* sehingga maknanya tetap terwakili meskipun tidak muncul dalam kamus standar. Sementara itu, *XLMRoBERTaTokenizer* yang menggunakan pendekatan *SentencePiece* dipilih karena lebih adaptif terhadap teks yang bersifat *noisy* dan tidak terstruktur, terutama dari media sosial, tanpa perlu proses pra-tokenisasi berbasis spasi. Pendekatan ini membantu menjaga konteks dan konsistensi pemrosesan data selama tahap tokenisasi.

1. *BERTTokenizer* : Penggunaan *BERTTokenizer* dengan algoritma *WordPiece* pada *IndoBERT* dan *IndoBERTweet* membantu sistem memahami karakter Bahasa Indonesia yang bersifat aglutinatif serta penuh variasi bahasa informal dalam dataset. *WordPiece* bekerja dengan memecah kata yang panjang atau kompleks menjadi potongan *sub-word* yang tetap bermakna, sehingga model tetap bisa menangkap arti dasar dari kata berimbuhan dalam konteks politik. misalnya kata “mempolitisasi” dapat diurai tanpa kehilangan maknanya. Kesesuaian *tokenizer* ini juga diperkuat oleh karakter masing-masing model *IndoBERTweet* lebih akurat mengenali slang khas media sosial seperti “bkn” atau “yg”.
2. *XLMRoBERTaTokenizer* : *XLMRoBERTaTokenizer* yang menggunakan algoritma *SentencePiece* dipilih untuk model *XLM-RoBERTa* karena mampu

menghadapi karakter bahasa media sosial yang sangat berisik dan beragam. Tidak seperti *tokenizer* biasa yang mengandalkan spasi, *SentencePiece* membaca teks sebagai aliran karakter mentah, sehingga jauh lebih tahan terhadap bentuk penulisan tidak baku yang sering muncul dalam ekspresi emosional *black campaign*. Misalnya, kata seperti “curangggg” yang hurufnya diulang atau “gakbecus” yang ditulis tanpa spasi tetap bisa dipetakan ke makna dasarnya tanpa hilang konteks. Ditambah lagi, sifat multilingualnya membuat *tokenizer* ini mampu menangani campur kode dengan baik, sehingga konteks tetap terjaga ketika pengguna mencampurkan Bahasa Indonesia, Inggris, atau bahasa daerah dalam satu kalimat yang sangat umum di percakapan politik di media sosial.

### 3.4.5 Model

Dalam penelitian ini, analisis performa dilakukan dengan membandingkan tiga varian model berbasis *Transformer*. Pemilihan ketiga model ini didasarkan pada karakteristik linguistik data media sosial Twitter di Indonesia yang cenderung informal, memuat singkatan (*slang*), serta sering mengalami campur kode (*code-mixing*) dengan bahasa asing. Berikut adalah penjelasan model yang digunakan beserta justifikasi penggunaannya terhadap dataset penelitian:

1. *IndoBERT* : *IndoBERT* digunakan sebagai tolok ukur utama (*baseline*) untuk model yang berfokus pada Bahasa Indonesia. Karena dataset penelitian berisi teks berbahasa Indonesia, model ini dianggap lebih mampu memahami struktur sintaksis dan makna semantik bahasa tersebut secara lebih tepat dibandingkan model multilingual yang bersifat umum. Pemilihan *IndoBERT* dimaksudkan untuk menilai sejauh mana model yang dilatih menggunakan bahasa formal dapat memproses konten kampanye hitam yang kerap memadukan ragam bahasa formal dan semi-formal.
2. *IndoBERTweet* : Data pada penelitian ini diperoleh dari platform Twitter, yang memiliki karakteristik bahasa tersendiri, seperti penggunaan singkatan tidak baku (misalnya “yg”, “bkn”, “ga”), istilah slang, serta elemen khas seperti *hashtag* dan *mention*. *IndoBERTweet* dipilih karena kosakata dan pemodelan

konteksnya telah disesuaikan dengan ragam bahasa informal tersebut. Dalam konteks deteksi *black campaign*, narasi negatif kerap disampaikan menggunakan bahasa kasar atau *slang* untuk menghindari sistem moderasi otomatis, sehingga model ini diperkirakan lebih mampu mengenali dan memahami nuansa bahasa tersebut.

3. *XLM-RoBERTa* : Alasan utama pemilihan XLM-R adalah teknik tokenisasi yang digunakan, yaitu *SentencePiece*. Berbeda dengan *WordPiece* pada *BERT* yang cenderung menghasilkan token [UNK] ketika menemui kata asing, salah eja, atau bentuk yang tidak dikenali, *SentencePiece* bekerja langsung pada teks mentah sehingga jauh lebih tahan terhadap *noise* dan kesalahan penulisan yang umum ditemukan dalam *tweet* kampanye. Selain itu, kemampuan multilingual XLM-R menjadikannya lebih fleksibel dalam memproses *tweet* yang memadukan Bahasa Indonesia dengan bahasa lain. Hal ini relevan karena dataset penelitian ini juga memuat penggunaan bahasa daerah seperti Jawa dan Sunda, sehingga XLM-R diharapkan mampu menangani variasi bahasa tersebut dengan lebih baik.

### 3.4.6 *Fine-Tuning*

*Fine-tuning* dilakukan karena model *pre-trained* seperti *IndoBERT* dan XLM-R, meskipun telah dilatih pada korpus besar dan memahami struktur bahasa secara umum, belum memiliki kemampuan spesifik untuk mengenali pola linguistik khas fenomena *black campaign*. Keberhasilan proses ini sangat dipengaruhi oleh pemilihan *hyperparameter*. Dengan menentukan konfigurasi *hyperparameter* yang tepat, model dapat beradaptasi secara optimal pada domain *black campaign* tanpa kehilangan stabilitas maupun kemampuan generalisasi. Pada penelitian ini, tiga *hyperparameter* utama yang digunakan adalah *learning rate*, jumlah *epoch*, *batch size*, dan *weight decay* yang masing-masing memiliki peran penting dalam mengendalikan proses *fine-tuning* agar menghasilkan kinerja terbaik. Berikut merupakan penjelasan dari masing-masing *hyperparameter* tersebut:

1. *Learning Rate* : Nilai *learning rate* dipilih pada rentang kecil  $1e-3$  hingga  $2e-5$  karena model *Transformer* sangat sensitif terhadap perubahan bobot selama

proses *fine-tuning*. *Learning rate* sendiri merupakan ukuran seberapa besar langkah pembaruan bobot yang dilakukan model pada setiap iterasi pelatihan, sehingga nilai yang terlalu besar dapat membuat model kehilangan stabilitas, sedangkan nilai yang terlalu kecil membuat proses belajar menjadi lambat.

2. *Epoch* : Jumlah *epoch* dibatasi hingga 10, tetapi proses pelatihan sebenarnya dikontrol oleh *Early Stopping* dengan *patience* 5. Mekanisme ini mencegah *overfitting* karena model *Transformer* biasanya cepat beradaptasi, sehingga tidak perlu dilatih terlalu lama. Pelatihan otomatis dihentikan jika *validation loss* tidak membaik selama tiga *epoch* berturut-turut, dan *restore\_best\_weights* memastikan model yang dipakai untuk evaluasi adalah model dengan performa terbaik, bukan model di *epoch* terakhir yang mungkin sudah kehilangan kemampuan generalisasi.
3. *Batch Size* : Selain mempertimbangkan keterbatasan memori GPU, pemilihan *batch size* kecil seperti 8 dan 16 dilakukan karena ukuran *batch* ini jauh lebih membantu proses belajar model pada dataset yang tidak terlalu besar. *Batch size* sendiri adalah jumlah data yang diproses sekaligus dalam satu langkah pelatihan. Dengan *batch* yang kecil, model dapat memperbarui bobotnya lebih sering di setiap *epoch*, sehingga proses belajar menjadi lebih halus dan responsif terhadap variasi data. Berbeda dengan *batch size* besar yang kadang membuat model terlalu cepat puas dan melewatkan detail penting, penggunaan *batch size* kecil justru membantu model menemukan pola secara bertahap dan lebih stabil. Pendekatan ini membuat model lebih adaptif dan memiliki kemampuan generalisasi yang lebih baik terhadap data baru.
4. *Weight Decay* : Parameter *weight decay* sebesar 0.01 dan 0.001 digunakan sebagai bentuk regularisasi untuk mengurangi risiko *overfitting* pada model *Transformer* yang memiliki jumlah parameter sangat besar. *Weight decay* bekerja dengan memberikan penalti pada bobot yang berkembang terlalu besar selama pelatihan, sehingga model terdorong untuk mempertahankan bobot yang lebih kecil dan stabil. Pendekatan ini membantu model tidak sekadar menghafal *noise* pada data latih, tetapi benar-benar mempelajari pola yang lebih umum, sehingga performanya menjadi lebih baik dan lebih konsisten saat diuji pada data yang belum pernah dilihat.

5. *Early Stopping* : Penerapan mekanisme *Early Stopping* pada penelitian ini berfungsi sebagai strategi kontrol otomatis untuk mencegah *overfitting* sekaligus mengoptimalkan penggunaan sumber daya komputasi dengan memantau performa model secara *real-time* pada data validasi. Dengan menetapkan nilai *patience* sebesar 3, proses pelatihan akan dihentikan lebih awal apabila *validation loss* tidak mengalami peningkatan (penurunan nilai) selama tiga *epoch* berturut-turut. Mekanisme ini dipadukan dengan fitur *restore\_best\_weights*, yang memastikan bahwa model akhir yang disimpan adalah model dengan performa validasi terbaik yakni pada *epoch* dengan nilai *loss* terendah bukan model pada iterasi terakhir yang mungkin telah mengalami penurunan kemampuan generalisasi akibat pelatihan yang berlebihan.
6. *Hyperparameter Tuning* : Pemanfaatan subset data sebesar 1.500 sampel pada tahap *hyperparameter tuning* dilakukan dengan mempertimbangkan efisiensi komputasi dan percepatan waktu eksperimen tanpa mengurangi validitas statistik. Karena metode *grid search* mengharuskan model dilatih berulang kali untuk setiap kombinasi parameter yang diuji, penggunaan seluruh dataset akan menghasilkan beban komputasi yang sangat tinggi dan memerlukan waktu pelatihan yang kurang efisien. Dengan menerapkan teknik *stratified random sampling*, subset yang diambil tetap mempertahankan distribusi kelas yang proporsional dengan data asli, sehingga mampu memberikan estimasi yang reliabel terkait konfigurasi parameter terbaik meliputi *learning rate*, *batch size*, dan *weight decay* sebelum digunakan pada proses pelatihan penuh (*fine-tuning*).

### 3.4.7 Evaluasi Model

Evaluasi model dilakukan untuk mengukur sejauh mana model yang telah melalui proses *fine-tuning* mampu melakukan prediksi secara akurat terhadap data yang tidak pernah dilihat sebelumnya. Tahap ini penting karena tanpa evaluasi, performa model hanya akan terlihat pada data latih, sehingga berpotensi menimbulkan ilusi performa akibat *overfitting*. Evaluasi diperlukan untuk memastikan bahwa model benar-benar memahami pola *black campaign* secara umum, bukan sekadar menghafal contoh pada dataset. Dengan demikian, evaluasi model memberikan

gambaran objektif mengenai kualitas, keandalan, dan kemampuan generalisasi model dalam tugas klasifikasi *black campaign detection*.

Pada tahap evaluasi, model diuji menggunakan *data testing* yang telah dipisahkan sejak awal untuk menjamin objektivitas hasil dan di uji atau *testing model* menggunakan data eksternal untuk mengevaluasi seberapa baik model yang digunakan. Beberapa metrik evaluasi umum yang dihitung meliputi *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* memberikan gambaran umum tingkat prediksi benar secara keseluruhan, sedangkan *precision* mengukur ketepatan model dalam mendeteksi kelas *black campaign* tanpa menghasilkan terlalu banyak *false positive*. *Recall* digunakan untuk menilai kemampuan model menangkap seluruh kasus *black campaign*, dan *F1-score* digunakan sebagai metrik keseimbangan antara *precision* dan *recall*. Selain metrik numerik, sering pula dibuat *confusion matrix* untuk melihat distribusi kesalahan model secara lebih rinci.

### 3.4.8 Pengujian Generalisasi Model

Pengujian generalisasi model merupakan tahap lanjutan setelah proses pelatihan dan evaluasi model menggunakan data latih dan data uji dari dataset Pemilihan Presiden Indonesia 2024. Pada tahap ini, model yang telah melewati proses *fine-tuning* tidak mengalami proses pelatihan ulang, melainkan langsung diuji menggunakan dataset eksternal yang memiliki topik dan karakteristik berbeda dari data latih. Hal ini dilakukan untuk memastikan bahwa pengujian benar-benar merefleksikan kemampuan generalisasi model.

Dataset eksternal yang digunakan berasal dari luar domain pemilihan presiden dan disiapkan melalui proses konversi label agar sesuai dengan skema klasifikasi *black campaign* dan *non-black campaign*. Selanjutnya, model melakukan proses inferensi terhadap dataset tersebut dan hasil prediksi dievaluasi menggunakan metrik evaluasi. Hasil dari pengujian ini dianalisis untuk mengamati dampak perbedaan domain terhadap kinerja model serta untuk mengidentifikasi keterbatasan model dalam menangani data dengan konteks yang berbeda.

### **3.4.9 Analisis Performa**

Melalui analisis performa, peneliti dapat menilai efektivitas model pada kondisi nyata, mengidentifikasi aspek yang masih lemah, serta memberikan dasar yang kuat dalam menentukan model terbaik dan menarik kesimpulan bahwa pendekatan yang digunakan benar-benar relevan dan dapat diterapkan secara praktis.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan terkait analisis performa deteksi *black campaign* pada studi kasus Pemilihan Presiden Indonesia 2024 menggunakan model arsitektur *transformer* yaitu *IndoBERT*, *IndoBERTweet*, dan *XLM-RoBERTa*, diperoleh beberapa kesimpulan:

1. Penerapan model *Transformer-based* yang meliputi *preprocessing* teks, penyeimbangan data, dan proses *fine-tuning* efektif dalam mengklasifikasikan konten *black campaign* dan *non-black campaign* pada dataset Pemilihan Presiden Indonesia 2024.
2. Seluruh model *Transformer* menunjukkan performa yang sangat baik pada data uji internal, dengan akurasi di atas 93%. Model *IndoBERT* memberikan performa terbaik dengan nilai *accuracy* sebesar 94,45%, *precision* 0,9756, *recall* 0,9479, dan *F1-score* 0,9615, sehingga menjadi model paling optimal dalam penelitian ini.
3. Dibandingkan dengan model LSTM pada penelitian terdahulu, model *Transformer* menunjukkan peningkatan performa yang signifikan, menegaskan keunggulan *Transformer* dalam memahami konteks bahasa pada data politik yang kompleks.
4. Pengujian pada dataset eksternal menunjukkan penurunan performa yang cukup signifikan, yang mengindikasikan bahwa kemampuan generalisasi model masih terbatas ketika dihadapkan pada domain yang berbeda.

## 5.2 Saran

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

1. Penelitian selanjutnya disarankan menggunakan dataset yang lebih besar dan beragam untuk meningkatkan kemampuan generalisasi model.
2. Penelitian selanjutnya dapat mengembangkan atau mengombinasikan arsitektur *Transformer* guna memperoleh hasil yang lebih optimal.
3. Metode penyeimbangan data berbasis penambahan data sintetis, seperti SMOTE atau *data augmentation* teks, dapat dipertimbangkan untuk melihat pengaruhnya terhadap kinerja model klasifikasi.

## DAFTAR PUSTAKA

- Abidin, Z., Suratno, T., dan Putri, M. F. (2025). Penerapan Random Oversampling dan Principal Component Analysis untuk Meningkatkan Akurasi Prediksi Kebangkrutan Perusahaan di Indonesia dengan Model Machine Learning. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 12(5), 1209–1220. <https://doi.org/doi.org/10.25126/jtiik.2025125>
- Agustini, D., Firdaus, M. I., Farida, M., Rosadi, M. E., Noor, H., dan Muttaqin, R. (2025). Peningkatan Kinerja Named Entity Recognition Bahasa Indonesia melalui Augmentasi Data Berbasis Large Language Models. *Jurnal Informatika Teknologi Dan Sains (JINTEKS)*, 7(3), 1361–1369. <https://doi.org/10.51401/jinteks.v7i3.6403>
- Al-Faruq, dan Aziz, U. A. (2021). Implementasi Arsitektur Transformer pada Image Captioning dengan Bahasa Indonesia. *Journal Universitas Islam Indonesia*, 2(2021), 2. <https://journal.uui.ac.id/AUTOMATA/article/view/19525/11561>
- Aliferis, C., and Simon, G. (2024). *Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI*. [https://doi.org/10.1007/978-3-031-39355-6\\_10](https://doi.org/10.1007/978-3-031-39355-6_10)
- Anam, M. K., Defit, S., Efrizoni, L., and Firdaus, M. B. (2024). Early Stopping on CNN-LSTM Development to Improve Classification Performance. *Journal of Applied Data Sciences*, 5(3), 1175–1188. <https://doi.org/10.47738/jads.v5i3.312>
- Ardinata, P. M. S., Permana, A. A. J., dan Wijaya, I. N. S. W. (2024). Identifikasi dan Normalisasi Teks Slang dengan FastText pada Twitter dalam Bahasa

- Indonesia. *Jurnal Pendidikan Teknologi Dan Kejuruan*, 21(1).  
<https://ejournal.undiksha.ac.id/index.php/JPTK/article/download/66381/28533/213437>
- Awalina, A., Bachtiar, F. A., dan Utamingrum, F. (2022). Perbandingan Pretrained Model Transformer pada Deteksi Comparison of Pretrained Transformer Models on Spam Review. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 9(3). <https://doi.org/10.25126/jtiik.202295696>
- Azhari, M. R., dan Muhjad, H. (2024). Kampanye Hitam Pemilu pada Media Sosial. *Jurnal Pendidikan Tambusai*, 8(2), 31930–31938.  
<https://jptam.org/index.php/jptam/article/view/18211>
- Chandradev, V., Agus, I. M., Suarjaya, D., dan Bayupati, I. P. A. (2022). Analisis Sentimen Review Hotel Menggunakan Metode Deep Learning BERT. *Journal Buana Informatika JBI*, 14(2), 107–116.  
<https://doi.org/10.24002/jbi.v14i02.7244>
- Damayanti, N. M., Ariningtyas, I. D., Izuddin, M., dan Icham, A. (2025). Analisis Sentimen Publik Pada Tagar #BTSComeback Di Platform X Menggunakan Indobertweet. *JITET (Jurnal Informatika Dan Teknik Elektro Terapan)*, 13(3), 4–8. <https://doi.org/10.23960/jitet.v13i3.7176>
- Darmawan, I., Ramadhani, N., dan Dewi, N. P. (2025). Sistem Pemodelan T5 dalam Natural Language Processing untuk Memberikan Jawaban terhadap Sebuah Pernyataan Pengguna dalam Bahasa Indonesia. *Jurnal Kecerdasan Buatan , Komputasi Dan Teknologi Informasi*, 6(1), 1–10.  
<https://doi.org/10.33650/coreai.v6i1.11280>
- Firmanto, B., Aziz, S., dan Sesoca, J. (2024). Tinjauan Perkembangan Kecerdasan Buatan Berbasis Arsitektur Transformer. *SMARTICS Journal*, 10(1), 33–38.  
<http://ejournal.unikama.ac.id/index.php/jst/article/view/8351/4497>
- Gomstyn, A. (2024). *What is data validation?* International Business Machines Corporation. <https://www.ibm.com/id-id/think/topics/data-validation>. Diakses pada 9 November 2025.

- Heydarian, M., and Doyle, T. E. (2022). MLCM : Multi-Label Confusion Matrix. *Canadian Department of National Defence Innovation for Defence Excellence and Security (IDEaS)*, 10, 19083–19095. <https://doi.org/10.1109/ACCESS.2022.3151048>
- Holdsworth, J. (2024). *What is tokenization?* International Business Machines Corporation. <https://www.ibm.com/id-id/think/topics/tokenization>. Diakses pada 9 November 2025.
- Ilemobayo, J. A., Durodola, O., Ogungbire, A., and Osinuga, A. (2024). Hyperparameter Tuning in Machine Learning : A Comprehensive Review. *Journal of Engineering Research and Reports*, 26(6), 388–395. <https://doi.org/10.9734/jerr/2024/v26i61188>
- Jeconiah, C. (2025). Peran Media Digital Dalam Mempengaruhi Persepsi Publik Terhadap Kandidat Presiden 2024. *TRIWIKRAMA: Jurnal Multidisiplin Ilmu Sosial*, 7(6), 1. <https://ejournal.warunayama.org/index.php/triwikrama/article/view/11375>. Diakses pada 9 November 2025.
- Johannes, A. Z., Bukit, M., Betan, A. D., and Tarigan, J. (2024). Google Teachable Machine: Pemanfaatan Machine Learning Berbasis Cnn Untuk Identifikasi Cepat Batuan Mineral Kalsit, Kuarsa, Dan Magnetit. *J-Icon : Jurnal Informatika Dan Komputer*, 12(2), 102–109. <https://doi.org/10.35508/jicon.v12i2.15170>
- Kania, S., Alya, N., Jumiatin, I., dan Sulistia, I. (2025). Implementation of Hyperparameter Tuning for Classification Models in Heart Disease Risk Prediction Penerapan Hyperparameter Tuning pada Model Klasifikasi untuk Prediksi Risiko Penyakit Jantung. *Institut Riset Dan Publikasi Indonesia (IRPI)*, 5(4), 1181–1189. <https://doi.org/10.57152/malcom.v5i4.2138>
- Keenan Wiciaputra, Y. (2021). Implementasi XLM-RoBERTa untuk Klasifikasi Teks Berbahasa Inggris dan Indonesia. *Universitas Multimedia Nusantara Library*, 13(3), 8–15. <https://kc.umn.ac.id/id/eprint/17598>
- Khatavkar, V., Petkar, S., and Vaidya, A. S. (2025). Multilingual Transformer

- Contextual Embedding Model for Political Tweets Analysis. *Cureus Journal of Computer Science*, 2, 1–11. <https://doi.org/10.7759/s44389-025-03177-4>
- Komaran, R. M. (2025). Model Klasifikasi Sentimen Berita Online Menggunakan Metode TF-IDF+ SVM dan Fine-Tuned IndoBERT. *ULBI Repository*, 15–17. <https://eprints.ulbi.ac.id/3465/>
- Komisi Pemilihan Umum Kabupaten Yalimo. (2025). *Mengenal Black Campaign dalam Politik: Arti, Ciri, dan Contohnya*. KPU Kabupaten Yalimo Blog. [https://kab-yalimo.kpu.go.id/blog/read/8618\\_mengenal-black-campaign-dalam-politik-arti-ciri-dan-contohnya](https://kab-yalimo.kpu.go.id/blog/read/8618_mengenal-black-campaign-dalam-politik-arti-ciri-dan-contohnya). Diakses pada 6 November 2025.
- Lambruschini, B. C. (2023). Reducing tokenizer's tokens per word ratio in Financial domain with T-MuFin BERT Tokenizer. *Proceedings of the Joint Workshop of the 5th Financial Technology and Natural Language Processing (FinNLP) and 2nd Multimodal AI For Financial Forecasting (Muffin)*, 94–103. <https://aclanthology.org/2023.finnlp-1.9/>
- Maxwell, A. E., and Warner, T. A. (2021). Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies. *Remote Sensing Studies*, 13(3), 9. <https://doi.org/10.3390/rs13132450>
- Murel, J. (2024). *What is downsampling or sample reduction?* International Business Machines Corporation. <https://www.ibm.com/id-id/think/topics/downsampling>. Diakses pada 9 November 2025.
- Nabiilah, G. Z., Alam, I. N., Purwanto, E. S., dan Hidayat, M. F. (2024). Indonesian multilabel classification using IndoBERT embedding and MBERT classification. *International Journal of Electrical and Computer Engineering (IJECE)*, 14(1), 1071–1078. <https://doi.org/10.11591/ijece.v14i1.pp1071-1078>
- Nyandwi, J. (2023). *The Transformer Blueprint: A Holistic Guide to the Transformer Neural Network Architecture*. Deep Learning Revision. <https://deeprevision.github.io/posts/001-transformer/>
- Pan, L. (2020). Towards Understanding Neural Collapse : The Effects of Batch

- Normalization and Weight Decay. *ArXiv and ACL Anthology*, 1, 3–4. <https://doi.org/10.48550/arXiv.2309.04644>
- Paraijun, F., Aziza, R. N., dan Kuswardani, D. (2022). Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah. *KILAT: Jurnal Ilmiah Teknologi Dan Informasi*, 11(1), 1–9. <https://doi.org/10.33322/kilat.v10i2.1458>
- Petru, P. (2023). *Transformer Architecture: What Is a Transformer?* Roboflow Blog Inc. <https://blog.roboflow.com/what-is-a-transformer/>
- Pradana, A. I. (2024). Perbandingan Data Untuk Memprediksi Ketepatan Studi Berdasarkan Atribut Keluarga Menggunakan Machine Learning. *JIKA (Jurnal of Informatics) Universitas Muhammadiyah Tangerang*, 8(2), 221–228. <https://doi.org/10.31000/jika.v8i2.10752>
- Prasetyo, E. A. (2024). Chatbot untuk Informasi Pembangunan Wilayah Kota Semarang Menggunakan Metode Retrieval Augmented Generations (RAG). *Jurnal Teknologi Komputer Dan Informatika*, 3(2). <https://doi.org/10.59820/tekomin.v3i2.336>
- Pratama, A. R., Wabula, F., Imandry, H., Isabela, M. L., dan Sianipar, R. (2025). Literature Review The Impact of Machine Learning in Modern Industries. *Jurnal Ilmiah Mahasiswa*, 3(1), 177–182. <https://doi.org/10.59603/niantanasikka.v3i1.680>
- Priambodo, W. (2024). *Black Campaign Detection Dataset*. GitHub. <https://github.com/wisnupri/Black-Campaign-Detection>. Diakses pada 18 Januari 2026.
- Priambodo, W., dan Zuliarso, E. (2024). Kombinasi K-Means dan LSTM untuk Deteksi Black Campaign di Media Sosial pada Calon Presiden Indonesia 2024. *Jurnal Teknik Informatika (JUTIF)*, 5(2), 539–550. <https://doi.org/10.52436/1.jutif.2024.5.2.1635>
- Ramadhan, C., Atina, V., dan Permatasari, H. (2025). Analisis Perbandingan Model CNN dan IndoBERT Dalam Sentimen Berita Politik Indonesia. *Seminar*

- Nasional Teknologi Informasi Dan Bisnis (SENATIB) 2025*, 110–118.  
<https://doi.org/10.47701/v1r9ka69>
- Ramanathan, N., Sivanaiah, R., Suseelan, A. D., Thanka, M., and Thanagathai, N. (2023). TechSSN at SemEval-2023 Task 12: Monolingual Sentiment Classification in Hausa Tweets. *Conference: Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, 1190–1194.  
<https://doi.org/doi.org/10.18653/v1/2023.semeval-1.165>
- Salve, H. N., Tubil, P. N. T., Salve, H. N., and Tubil, P. N. T. (2025). Tokenization Efficiency in Code-Switched Text : Comparing SentencePiece and Byte-Pair Encoding on Taglish. *IEEE (Institute of Electrical and Electronics Engineers)*, 2–9. <https://doi.org/10.36227/techrxiv.175756338.86755792/v1>
- Suhaeni, C., Wijayanto, H., dan Kurnia, A. (2024). Sentiment Classification on the 2024 Indonesian Presidential Candidate Dataset Using Deep Learning Approaches. *Indonesian Journal of Statistics and Its Applications*, 8(2), 83–94. <https://doi.org/10.29244/ijsa.v8i2p83-94>
- Thanzani, A., Dean, A., Sari, P., Yulia, L. T., dan Fikri, S. (2022). Black Campaign Melalui Media Elektronik dari Perspektif Hukum Pemilu. *Journal Evidence Of Law*, 1(3), 42–51. <https://doi.org/10.59066/jel.v1i3.103>
- Vaswani, A. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30 (NIPS)*, 30, 5998–6008.
- Widhi, I. M. (2025). Analisis Klasifikasi Tweet Berdasarkan Topik Sosial Menggunakan SVM. *Jurnal Nasional Teknologi Informasi Dan Aplikasinya JNATIA*, 3(4), 855–864.  
<https://doi.org/https://doi.org10.24843/JNATIA.2025.v03.i04.p15>
- Wijaya, H., dan Hayati, N. (2025). Natural Language Processing (NLP) for Sentiment Analysis of Seblak Bandung Pedas Kudus Reviews. *Jurnal of Business and Audit Information System (JBASE)*, 8(1), 13–22.  
<https://doi.org/10.30813/jbase.v8i1.8035>
- Yulianti, E., dan Nissa, N. K. (2024). ABSA of Indonesian customer reviews using

IndoBERT : single- sentence and sentence-pair classification approaches.  
*Bulletin of Electrical Engineering and Informatics*, 13(5), 3579–3589.  
<https://doi.org/10.11591/eei.v13i5.8032>