BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

Pada bagian landasan teori ini dibahas konsep dan metode yang menjadi dasar dalam penelitian deteksi komentar *toxic* dalam skenario *cross-lingual* antara bahasa Indonesia dan bahasa Inggris.

2.1.1. Media Sosial

Media sosial didefinisikan sebagai kumpulan aplikasi berbasis internet yang dibangun di atas fondasi web 2.0 dan memungkinkan terciptanya serta pertukaran konten yang dihasilkan oleh pengguna (Kaplan & Haenlein, 2010). Media sosial tidak terbatas pada jejaring sosial populer seperti X dan Facebook saja tetapi juga mencakup berbagai bentuk platform interaksi daring lain seperti halaman diskusi Wikipedia hingga kolom komentar berita selama platform tersebut memfasilitasi interaksi sosial (Kaplan & Haenlein, 2010). Dalam kajiannya, (Modi & Patel, 2025) menjelaskan bahwa karakteristik utama media sosial terletak pada *user-generated content,* interaktivitas tinggi, dan juga domain yang sangat luas membedakannya dari media tradisional.

Namun, karakteristik yang sama juga membuka ruang bagi munculnya perilaku komunikasi yang menyimpang. (Schmitt dkk., 2025) menjelaskan fenomena *toxic online disinhibition*, yaitu kondisi ketika individu cenderung mengekspresikan perilaku yang lebih agresif, kasar, atau tidak terkendali dalam interaksi daring dibandingkan interaksi tatap muka yang disebabkan faktor seperti anonimitas dan jarak antar pengguna dalam media sosial mendorong menurunnya

kontrol diri pengguna. Akibatnya, perilaku menyimpang lebih mudah muncul dan menimbulkan dampak negatif, terutama terhadap kondisi psikologis korban.

2.1.2. Natural Language Processing (NLP)

Natural Language Processing (NLP) atau pemrosesan bahasa alami merupakan bidang dalam kecerdasan buatan yang berfokus pada kemampuan komputer untuk memahami, memproses dan menghasilkan bahasa manusia. Menurut (Jurafsky & Martin, 2008), NLP bertujuan agar memungkinkan komputer melakukan pemrosesan bahasa alami sehingga dapat berinteraksi dengan manusia menggunakan bahasa yang digunakan sehari-hari. NLP saat ini banyak digunakan dalam berbagai aplikasi seperti mesin pencari, sistem terjemahan bahasa otomatis, analisis sentimen serta sistem rekomendasi (Aditi Pancholi, 2021).

1. Analisis Sentimen

Analisis sentimen merupakan salah satu subbidang dari NLP yang berfokus pada proses identifikasi dan klasifikasi opini atau emosi yang terkandung dalam teks ke dalam kategori tertentu. Analisis sentimen bertujuan untuk memahami sikap, pendapat, serta persepsi pengguna terhadap suatu objek melalui data tekstual yang tersedia secara luas di internet (Tan dkk., 2023). Deteksi komentar *toxic* merupakan salah satu bentuk pengembangan dari analisis sentimen, di mana proses klasifikasi tidak lagi berfokus pada polaritas sentimen, tetapi pada identifikasi penggunaan bahasa kasar yang mengandung unsur ofensif dalam teks (Song, Huang, & Xiao, 2021).

2. *Toxic Comment*

Toxic comment pada media sosial dapat dipahami sebagai bentuk ujaran daring yang bersifat merugikan dan mencakup berbagai bentuk perilaku berbahasa. (Grotti, 2024) mendefinisikan *toxic language* sebagai istilah yang mencakup berbagai bentuk ujaran bermasalah dan sifatnya merugikan seperti *offensive* dan *abusive language, harassment, cyberbullying,* hingga ujaran cabul, *obscenity* dan *hate speech,* termasuk ekspresi yang tidak selalu dapat diproses secara hukum tetapi tetap berdampak negatif secara sosial.

Seiring berkembangnya media sosial dan meningkatnya intensitas komentar yang dipublikasikan, potensi munculnya ujaran *toxic* ikut membesar. *"However, publicly shared comments may also contain toxic content that can damage an individual's reputation"* (Shahid dkk., 2024) menegaskan bahwa komentar yang dibagikan secara publik tidak hanya menjadi sarana interaksi, tetapi juga media penyebaran dampak sosial yang merugikan.

Pada penelitian ini, komentar *toxic* yang digunakan mencakup enam kategori, yaitu *toxic*, *severe toxic*, *obscene*, *insult*, *threat*, dan *identity hate*, yang kemudian digabungkan menjadi satu kelas yaitu *toxic*. Berikut disajikan contoh dari masing-masing jenis komentar tersebut.

Tabel 2. 1 Contoh komentar *toxic*

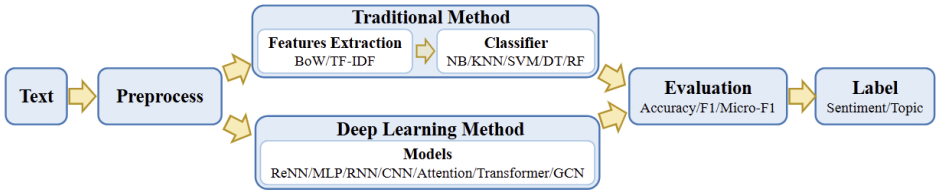
Komentar	**Bahasa**	**Label**
*"USER B*jingan. \nGw gak peduli aksi-aksi kemaren. Katanya murni panggilan modal sendiri.\nBalikin duit Gw b*jingan. \nGegayaan*	Indonesian	*toxic*

Komentar	Bahasa	Label
panggilan Allah. Modal sendiri, ternyata make duit jamaah."		
*"KPK banyak semrawut juga g*blok...bubarkan saja c USER toh banyak kasus mengambang begitu saja..hahh h*ram URL"*	Indonesian	*severe_toxic*
*"YOU GUYS ALL S*CK, LET ME EDIT!!! \n IM AWESOME!!!"*	English	*obscene*
*"Let us please hunt this bi*ch down and k*ll it, like we should have been allowed to do with bernardo, homolka and olsen."*	English	*threat*
*"waste of oxygen \n\nEven amongst je*koff wikilos*r admins who have no life, you are a sad sack. Seriously, have ever considered su*cide?"*	English	*insult*
"Rezim ini bener2 dikuasai oleh oknum2 yg anti-Islam"	Indonesian	*identity_hate*
"\n\n Barnstar \n\n The Press Barnstar Thanks for giving the interview! I enjoyed the read. Thanks for your contributions. Best wishes."	English	*neutral*

2.1.3. Text Classifier

Text Classifier merupakan metode yang digunakan untuk mengelompokkan teks ke dalam kategori tertentu berdasarkan isi teks. Sementara itu *Text Classification* mengacu pada proses ekstraksi fitur dari data teks mentah dan memprediksi kategori teks berdasarkan fitur-fitur tersebut. Di era ledakan informasi, klasifikasi teks secara manual menjadi tidak efisien dan subjektif, sehingga diperlukan model otomasi berbasis *machine learning* dan *deep learning*

untuk melakukan klasifikasi secara cepat, konsisten, dan objektif (Qian Li dkk., 2021).



Gambar 2. 1 *Flowchart Text Classifier* (Qian Li dkk., 2021)

Gambar 2.1 menunjukkan *flowchart* metode-metode *text classification,* di mana ekstraksi fitur merupakan tahap penting dalam proses klasifikasi teks, pada metode tradisional ekstraksi fitur dilakukan secara terpisah, sedangkan pada *deep learning* proses tersebut dilakukan secara otomatis oleh model.

(Tidake & Sane, 2018) membedakan jenis-jenis *text classification* berdasarkan karakteristik label sebagai berikut:

1. *Single-label classification*

Pendekatan di mana setiap teks hanya diberikan satu label yang paling merepresentasikan isi teks.

2. *Binary classification*

Merupakan bentuk khusus dari *single-label classification* dengan dua kelas yang saling berlawanan seperti positif dan negatif atau *toxic* dan *non-toxic.*

3. *Multi-class classification*

Klasifikasi teks ke dalam satu dari lebih dari dua kelas yang saling eksklusif.

4. *Multi-label classification*

Pendekatan yang memungkinkan satu teks memiliki lebih dari satu label secara bersamaan karena teks dapat merepresentasikan beberapa kategori.

2.1.4. Fine-tuning

Fine-tuning merupakan proses penyesuaian parameter pada model kecerdasan buatan yang sudah dilatih sebelumnya agar representasi pengetahuan model dapat beradaptasi terhadap tugas yang lebih spesifik (Sun dkk., 2020). Perbedaan mendasarnya dengan metode ekstraksi fitur yaitu teknisnya yang melakukan *unfreeze* sebagian atau seluruh *layer* model sehingga bobot model ikut diperbaharui (Peters dkk., 2019).

Berbagai studi empiris menunjukkan bahwa pendekatan *fine-tuning* umumnya cenderung menghasilkan performa yang lebih baik dibandingkan metode ekstraksi fitur (Peters dkk., 2019), khususnya pada tugas klasifikasi teks menggunakan arsitektur berbasis BERT (Sun dkk., 2020).

Meskipun begitu, proses *fine-tuning* harus dilakukan dengan hati-hati karena metode *fine-tuning* yang terlalu agresif berpotensi menyebabkan *catasthropic forgetting* pada model dasar yaitu kondisi di mana model melupakan pengetahuan *universal* yang telah di pelajari pada fase pelatihan sebelumnya (Howard & Ruder, 2018). Pada arsitektur berbasis BERT, modifikasi parameter yang terlalu agresif pada layer bawah sangat rentan memicu kondisi ini (Sun dkk., 2020).

Namun jika diterapkan dengan benar, metode *fine-tuning* bahkan dapat mengatasi permasalahan *class imbalance.* Penelitian (Valizadehaslani dkk., 2022) menemukan bahwa *fine-tuning* yang dilakukan dengan dua tahap dapat meningkatkan *f1-score* kelas minoritas sebesar 1,6% dibandingkan *vanilla fine-tuning.*

2.1.5. Multi-source Data

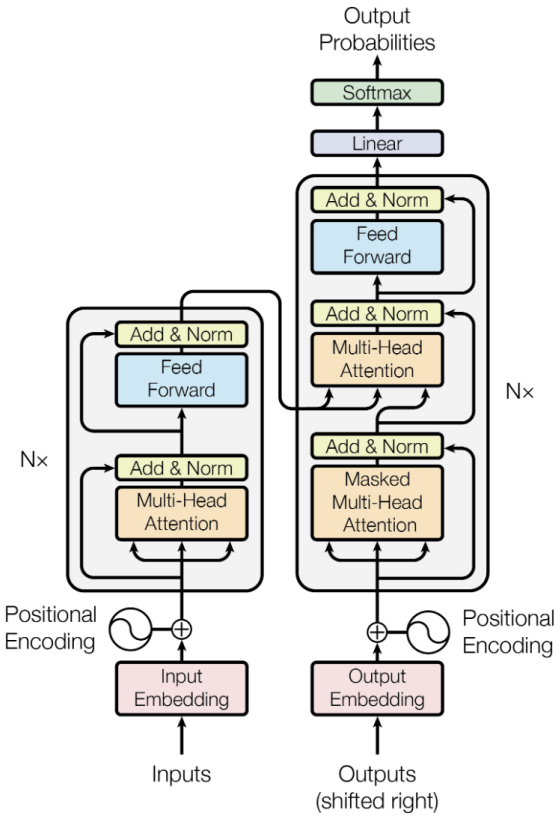
Multi-source data merupakan integrasi satu kesatuan teks dari berbagai sumber berbeda untuk memperoleh informasi yang lebih kaya (Han, 2025). *Multi-source data* memiliki kelebihan tersendiri dibandingkan data yang di ambil dari satu sumber, (Guo & Xing, 2025) menyatakan bahwa metode yang hanya bergantung pada satu sumber cenderung tidak mampu menangkap asosiasi semantik lintas sumber dan konteks yang lebih kaya secara utuh.

Pemanfaatan *multi-source data* sendiri telah diterapkan pada berbagai tugas, seperti penambangan pola perilaku tersembunyi dari interaksi media sosial lintas platform (Oliveira, 2025). Dalam konteks *toxic comment detection,* menggabungkan data dari banyak sumber membantu model memahami variasi ekspresi toksisitas yang lebih kaya dan meningkatkan generalisasi lintas domain.

Penelitian ini menggunakan dataset Jigsaw *Multilingual Toxic Comment Classification* (Kivlichan dkk., 2020) untuk bahasa Inggris dan Aldon Dataset untuk bahasa Indonesia. Jigsaw Dataset dikembangkan oleh Conversation AI milik Google dengan menghimpun komentar publik dari dua sumber yaitu *Civil Comments* yang di ambil dari situs berita dan Wikipedia Talk Pages*,* sementara itu, Aldon Dataset dikumpulkan dari platform media sosial Twitter atau X. Dengan demikian, konsep *multi-source* dalam penelitian ini berangkat dari keberagaman asal komentar publik lintas platform daring, di mana kedua dataset sama-sama merepresentasikan interaksi sosial digital dan memiliki karakteristik label yang serupa.

2.1.6. Transformer

Transformer pertama kali diperkenalkan oleh (Vaswani dkk., 2017) sebagai arsitektur berbasis *self-attention* yang sepenuhnya menghilangkan mekanisme rekuren dan konvolusi dalam pemodelan sekuens. Mekanisme *self-attention* memungkinkan setiap *token* dalam sekuens memperhatikan seluruh *token* lainnya secara langsung sehingga model dapat menangkap hubungan antar kata dalam seluruh konteks kalimat secara bersamaan.



Gambar 2. 2 Arsitektur Transformer (Vaswani dkk., 2017)

Arsitektur dasar transformer terdiri dari *encoder* di bagian kiri dan *decoder* dibagian kanan dengan komponen utama berupa *multi-head attention* dan *feed-*

forward network, yang dilengkapi *residual connection* dan *layer normalization.* Berikut penjelasan setiap komponen-komponennya.

1. *Multi-Head Attention*

Mekanisme utama yang menangkap hubungan antar *token* dalam satu sekuen.

2. *Masked Multi-Head Attention*

Versi *attention* pada *decoder* yang hanya melihat *token* sebelumnya (tidak boleh melihat masa depan).

3. *Feed Forward Network (FFN)*

Layer *fully connected* yang memproses fitur hasil *attention* untuk memperkaya representasi setiap katanya.

4. *Add & Norm*

Residual connection dan *Layer Normalization* untuk menjaga stabilitas dan mempertahankan representasi awal.

5. *Encoder-Decoder Attention*

Menghubungkan representasi *encoder* ke *decoder* saat menghasilkan *output*. *Encoder* bersifat memahami sementara *decoder* bersifat memprediksi.

6. *Linear, Softmax* dan *Nx Layer*

Linear layer memetakan representasi akhir menjadi skor mentah atau logit, *softmax* mengubah logit menjadi probabilitas dan *Nx Layer* merupakan blok *encoder* atau *decoder* yang diulang beberapa kali untuk memperdalam representasi.

Seiring perkembangannya, (T. Lin dkk., 2021) menjelaskan bahwa arsitektur transformer kemudian melahirkan banyak varian yang dimodifikasi dari tiga aspek utama, yaitu arsitektur, strategi *pre-training* dan pengaplikasian. Selain itu,

transformer juga menjadi pondasi lahirnya *Transformer-based Pretrained Models* seperti *Generative Pretrained Transformers* (GPT) dan BERT (Kalyan dkk., 2021). Di sisi lain, (Minaee dkk., 2024) menemukan bahwa kebanyakan *Large Language Model* merupakan *transformer-based neural language models* dengan skala parameter yang sangat besar, sehingga arsitektur transformer kini menjadi *backbone* dominan yang digunakan oleh mayoritas *Large Language Model* (LLM) terkini.

2.1.7. Robustly Optimized BERT Pretraining Approach

Robustly Optimized BERT Pretraining Approach (RoBERTa) merupakan pengembangan dari model *Bidirectional Encoder Representation from Transformer* (BERT) yang berfokus pada optimalisasi proses *pre-training* tanpa mengubah arsitektur nya secara mendasar (Y. Liu dkk., 2019).

BERT sendiri diperkenalkan sebagai *pre-trained deep bidirectional transformer encoder* yang dikembangkan melalui dua tahap yaitu *pre-training* untuk mempelajari representasi bahasa umum dari korpus besar dan *fine-tuning* untuk mengadaptasi model pada tugas spesifik. Pada tahap *pre-training,* BERT menggunakan *Masked Language Modeling* (MLM) untuk memprediksi *token* yang di tutup secara acak serta *Next Sentence Prediction* (NSP) untuk memahami hubungan antar kalimat. Secara arsitektur, BERT menggunakan Transformer *encoder* dengan dua konfigurasi utama yaitu BERTBASE dengan 110 juta parameter dan BERTLARGE dengan 340 juta parameter (Devlin dkk., 2019).

RoBERTa mempertahankan arsitektur Transformer *encoder* yang sama, tetapi melakukan beberapa perubahan penting pada strategi pelatihan dengan

menghapus NSP, menggunakan *dynamic masking* alih-alih *static masking,* serta memperluas data *pre-training* hingga lebih dari 160GB teks (Y. Liu dkk., 2019). RoBERTa juga hadir dalam skala besar seperti RoBERTaBASE dengan sekitar 125 juta parameter dan RoBERTaLARGE dengan sekitar 355 juta parameter sehingga memiliki kapasitas representasi lebih tinggi dibanding BERT standar.

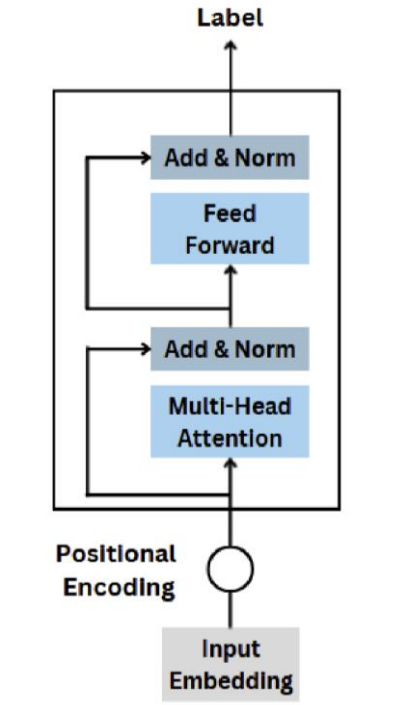
RoBERTa dikategorikan sebagai salah satu *transformer-based model* yang secara konsisten menunjukkan performa unggul pada berbagai tugas NLP, seperti analisis sentimen, *Named Entity Recognition* (NER), *question answering* dan tugas lainnya (Zhang & Shafiq, 2024). Keunggulan ini menjadikan RoBERTa sering digunakan sebagai *baseline* kuat dalam penelitian lanjutan.

2.1.8. Cross-lingual Language Model with RoBERTa Architecture

Cross-lingual Language Model with RoBERTa Architecture (XLM-RoBERTa) merupakan *cross-lingual language model* berbasis arsitektur model RoBERTa dengan menggunakan transformer *encoder* yang dikembangkan melalui *unsupervised multilingual masked language modeling* pada skala besar yaitu dilatih pada lebih dari 2TB data *Common Crawl* yang mencakup 100 bahasa termasuk bahasa Indonesia (Conneau dkk., 2020), seperti ditunjukkan pada Gambar 2.4.

XLM-RoBERTa hadir dalam dua varian, yaitu XLM-RoBERTa-base dengan jumlah parameter sekitar 278 juta dan XLM-RoBERTa-large dengan jumlah parameter sekitar 550 juta. Jumlah parameter yang lebih besar dibandingkan RoBERTa disebabkan oleh penggunaan *vocabulary multilingual* yang memungkinkan model mempelajari representasi bahasa dari berbagai bahasa secara bersamaan (Conneau dkk., 2020). Penelitian ini menggunakan XLM-RoBERTa-

base karena memiliki jumlah parameter yang lebih efisien, sehingga lebih sesuai untuk proses *fine-tuning* pada dataset dengan ukuran terbatas serta kebutuhan komputasi yang lebih ringan.



Gambar 2. 3 Arsitektur Transformer *encoder-only* (Dinarta & Wicaksana, 2025)

Model ini dibangun dengan *backbone* Transformer *encoder-only* yang terdiri atas *multi-head self-attention* dan *feed-forward network* yang ditumpuk sebanyak *12 layer,* tanpa mekanisme *decoder* dan *mask attention* (Dinarta & Wicaksana, 2025). Struktur ini yang kemudian menjadi *backbone* utama model BERT, RoBERTa dan XLM-RoBERTa. Berikut penjelasan dari setiap komponen pada arsitektur transformer *encoder.*

1. Multi-Head Self-Attention

Mekanisme *self-attention* memungkinkan setiap *token* menghitung keterkaitannya dengan seluruh *token* lain secara paralel dalam satu sekuen, sementara *multi-head*

self-attention menjalankan mekanisme tersebut secara paralel dalam beberapa *head* untuk menangkap beragam pola relasi dengan lebih kaya lalu menggabungkan setiap hasilnya.

2. Add & Normalization

Residual Connection dan normalisasi diterapkan untuk menjaga stabilitas pelatihan dan mempertahankan informasi awal.

3. Feed Forward Network

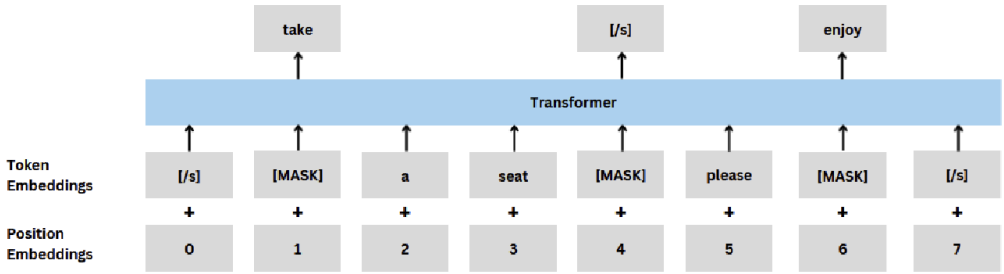
Layer *fully connected* yang memproses fitur hasil *attention* untuk memperkaya representasi setiap katanya.

4. Stacked Encoder Layers (N layers)

Blok *encoder* diulang beberapa kali untuk menghasilkan representasi yang semakin abstrak dan kompleks

5. Output Representation

Encoder menghasilkan representasi kontekstual yang dapat digunakan tugas *downstream* seperti klasifikasi.



Gambar 2. 4 Arsitektur MLM pada XLM-RoBERTa (Dinarta & Wicaksana, 2025)

Pada *Masked Language Modeling* (MLM), sebagian *token* pada input diganti dengan [MASK] dan model dilatih untuk memprediksi *token* asli berdasarkan konteks dua arah. Berikut penjelasan setiap komponen nya.

1. Token Embedding

Mengubah setiap *token* menjadi vektor numerik berdimensi tetap sebagai representasi awal makna kata.

2. Position Embedding

Ditambahkan ke *token embedding* untuk memberikan informasi urutan kata karena Transformer tidak memiliki mekanisme sekuensial bawaan.

3. Special Token [s] dan [/s]

[s] menandai awal sekuens (representasi global kalimat), sedangkan [/s] menandai akhir sekuens.

4. Token [MASK]

Sebagian *token* diganti dengan [MASK] untuk dilatih agar model memprediksi *token* asli menggunakan konteks sekitarnya.

5. Transformer Encoder

Terdiri dari *multi-head self-attention* dan *feed forward network* yang ditumpuk *N layers* untuk membangun representasi kontekstual dua arah.

6. Output Prediction Layer

Menghasilkan probabilitas *token* pada posisi [MASK] dan digunakan untuk menghitung *loss* selama *pre-training.*

XLM-RoBERTa dikembangkan untuk memungkinkan *transfer* pengetahuan dari bahasa *high-resource* ke bahasa *low-resource* melalui ruang representasi bersama (Xu dkk., 2024). Penelitian (Waysi Naaman dkk., 2025) juga menegaskan bahwa XLM-RoBERTa menjadi model *cross-lingual* yang dominan dari sisi performa karena *pre-training* berskala besar membuatnya unggul pada *zero-shot transfer* dan mampu memperkecil kesenjangan performa antara bahasa *high-resource* dan *low-resource.*

2.1.9. Imbalance Data Handling

Data imbalance merupakan kondisi ketika distribusi jumlah data antar kelas tidak seimbang, di mana satu kelas memiliki representasi jauh lebih besar dibanding kelas lainnya (He & Garcia, 2009). Kondisi ini menjadi masalah serius karena sebagian besar algoritma klasifikasi pada dasarnya diasumsikan bekerja pada distribusi data yang relatif seimbang. Dampaknya tidak hanya berupa penurunan performa model secara umum, tetapi juga meningkatnya risiko kesalahan klasifikasi pada kelas minoritas (Thabtah dkk., 2020). Oleh karena itu, penanganan khusus terhadap *imbalance data* menjadi penting agar model tidak hanya mengejar akurasi keseluruhan, tetapi juga mampu mengenali pola pada kelas minoritas secara seimbang (Hasanin dkk., 2019).

(Yuan dkk., 2023) mengelompokkan *imbalance data handling method* sebagai berikut:

1. Data-level method

Pendekatan yang bekerja pada tahap *preprocessing* dengan memodifikasi distribusi data, misalnya melalui *undersampling, oversampling,* atau teknik sintetik seperti

Synthetic Minority Oversampling Technique (SMOTE) untuk menyeimbangkan jumlah sampel antar kelas.

2. Algorithm-level method

Pendekatan yang tidak mengubah data, tetapi memodifikasi mekanisme pembelajaran model, misal melalui perubahan fungsi *loss* agar model lebih sensitif terhadap kelas minoritas.

3. Hybrid method

Pendekatan kombinasi yang menggabungkan strategi *data-level* dan *algorithm-level* untuk memperoleh performa yang lebih stabil pada data *imbalance.*

Dalam penelitian ini, penanganan *imbalance data* difokuskan pada *algorithm-level method*, karena tujuan utama penelitian adalah mengoptimalkan proses pembelajaran model melalui modifikasi fungsi *loss* dan strategi *fine-tuning* tanpa mengubah distribusi data. Pendekatan ini dipilih mengingat keterbatasan ketersediaan data asli pada bahasa dengan *low-resource language*. Metode *data-level* tidak diterapkan karena *oversampling* tidak memungkinkan akibat keterbatasan waktu dan tenaga dalam pengumpulan data tambahan (Yuan dkk., 2023). Sementara itu, *downsampling* juga tidak digunakan karena penghapusan sebagian data pada kelas mayoritas berpotensi menghilangkan pengetahuan dan informasi penting dari dataset (He & Garcia, 2009).

Tugas *toxic comment detection* pada penelitian ini merupakan *binary classification,* maka dari itu digunakan *Binary Cross-Entropy (BCE) loss* untuk mengukur perbedaan antara prediksi model dan label aktual, dengan Persamaan 1 (Song, Huang, & Xiao, 2021).

$$L_{BCE} = -\frac{1}{m}\sum_{i=1}^{m}(y_{(i)}\log(\hat{y}_{(i)}) + (1 - y_{(i)})\log(1 - \hat{y}_{(i)})) \quad (1)$$

Keterangan:

L_{BCE} : Nilai rata-rata *loss* BCE yang mengukur kesalahan prediksi model pada klasifikasi biner.

m : Jumlah total data (*sample*) dalam satu *batch* atau dataset yang dihitung nilai *loss*-nya.

$\sum_{i=1}^{m}()$: Operasi penjumlahan seluruh nilai *loss* dari data ke-1 sampai ke-m.

(i) : Indeks data ke-i dalam dataset atau *batch*.

$y_{(i)}$ *: Ground truth* data ke-i, bernilai 0 (negatif) atau 1 (positif).

$\hat{y}_{(i)}$: Probabilitas hasil prediksi model untuk data ke-i, bernilai antara 0 dan 1.

$\log(\hat{y}_{(i)})$: Logaritma natural dari probabilitas prediksi ketika label asli bernilai 1.

$\log(1 - \hat{y}_{(i)})$: Logaritma natural dari probabilitas prediksi ketika label asli bernilai 0.

$(y_{(i)}\log(\hat{y}_{(i)})$: Komponen *loss* yang aktif ketika label asli bernilai 1 (positif).

$\log(\hat{y}_{(i)}) + (1 - y_{(i)})\log(1 - \hat{y}_{(i)}))$: Komponen *loss* yang aktif ketika label asli bernilai 0 (negatif).

Tanda negatif $(-)$: Digunakan agar nilai *loss* menjadi positif karena log probabilitas bernilai ≤ 0.

$\frac{1}{m}$: Digunakan untuk menghitung rata-rata *loss* dari seluruh data.

Pada kondisi data *imbalance, loss* standar cenderung didominasi oleh sampel yang mudah diklasifikasikan sehingga pembelajaran kelas *minoritas* menjadi kurang optimal. *Focal loss* diperkenalkan untuk menurunkan kontribusi

easy samples dan memfokuskan *training* pada *hard samples* melalui faktor *modulating* $(1-p_t)^{\gamma}$, dengan Persamaan 2 (T.-Y. Lin dkk., 2018).

$$L_{FL} = -\frac{1}{m}\sum_{i=1}^{m}(y_{(i)}\alpha(1-\hat{y}_{(i)})^{\gamma}\log(\hat{y}_{(i)}) + (1-y_{(i)})\,(1-\alpha)\hat{y}_{(i)}^{\gamma}\log(1-\hat{y}_{(i)})) \quad (2)$$

Keterangan:

L_{FL} : Nilai rata-rata *loss* untuk klasifikasi biner dengan fokus pada *hard samples*.

α (alpha) : Parameter *class weight* untuk mengatasi ketidakseimbangan data.

$1-\alpha$: Bobot untuk kelas selain kelas yang diberi bobot α.

γ (gamma) : *Focusing parameter* yang mengurangi kontribusi *loss* dari sampel yang mudah diklasifikasikan dan memperbesar fokus pada sampel sulit.

$(1-\hat{y}_{(i)})^{\gamma}$: Faktor modulator untuk kelas positif yang memperkecil *loss* ketika prediksi sudah benar dengan probabilitas tertinggi.

$\hat{y}_{(i)}^{\gamma}$: Faktor modulator untuk kelas negatif yang memperkecil *loss* ketika prediksi sudah benar dengan probabilitas rendah (mendekati 0 untuk kelas positif).

Kedua *loss* digabungkan dalam rasio statis 1:1 agar model memperoleh keseimbangan antara stabilitas global dan fokus pada sampel minoritas. Fusi BCE dan Focal Loss ini digunakan sebagai *baseline* penelitian ini, dengan Persamaan 3 (Song, Huang, & Xiao, 2021).

$$L_{static} = \omega_1 * L_{BCE} + \omega_2 * L_{FL} \quad (3)$$

Penelitian (X. Liu dkk., 2024) mengusulkan *Dynamic-Recall Focal Loss* (DRFL) sebagai pengembangan dari *Focal Loss* dengan menambahkan komponen *class recall* agar *loss* menjadi lebih sensitif terhadap kelas minoritas pada data tidak seimbang. DRFL memodifikasi *modulation factor loss* dari $(1-p_t)^{\gamma}$ menjadi

$(1 - recall)^{\beta}(1 - p_t)^{\gamma}$ sehingga bobot *loss* meningkat ketika *recall* kelas rendah dan secara langsung mendorong peningkatan performa minoritas, dengan hasil eksperimen menunjukkan DRFL mampu melampaui Focal Loss lebih dari 2% pada metrik akurasi dan F1, dengan Persamaan 4.

$$L_{DRFL} = -(1 - recall)^{\beta} L_{FL} \quad (4)$$

Keterangan:

L_{DRFL} : Nilai *loss* yang menggabungkan Focal Loss dengan faktor berbasis *recall* untuk meningkatkan deteksi kelas minoritas.

L_{FL} : Focal Loss.

$recall$: Nilai *recall* kelas yang menunjukkan kemampuan model mendeteksi seluruh data minoritas.

$(1 - recall)^{\beta}$: Faktor penalti berbasis performa *recall,* semakin rendah *recall*, semakin besar nilai *loss*.

β : Parameter yang mengatur seberapa kuat pengaruh *recall* terhadap *loss*.

Penelitian ini menambahkan *adaptive loss weighting* yang terinspirasi dari metode SoftAdapt (Heydari dkk., 2019) sebagai pengembangan *static fusion loss* untuk menyesuaikan bobot *loss* secara dinamis berdasarkan performa setiap *loss* selama pelatihan. Mekanisme ini bekerja melalui tiga tahapan perhitungan, yaitu menghitung laju perubahan *loss* dengan persamaan 5, menghitung bobot adaptif menggunakan *softmax* dengan persamaan 6 dan menggabungkan bobot dari setiap *loss* dengan persamaan 7.

$$s_i^{(t)} = \frac{L_i^{(t)} - L_i^{(t-1)}}{L_i^{(t-1)} + \epsilon} \quad (5)$$

$$\omega_i{}^{(t)} = \frac{e^{\beta * s_i^{(t)}}}{\sum_j e^{\beta * s_j^{(t)}}} \quad (6)$$

$$L_{adaptive} = \omega_1^{(t)} * L_{BCE} + \omega_2^{(t)} * L_{FL/DRFL} \quad (7)$$

Keterangan:

$L_{adaptive}$: *Adaptive fusion loss.*

$s_i^{(t)}$: Laju perubahan komponen *loss* antar *batch.*

$\omega_i{}^{(t)}$: Bobot komponen *loss.*

(t) : *Batch.*

i, j : Komponen *loss* (BCE, Focal Loss, DRFL).

$L_i^{(t-1)} + \epsilon$: Faktor normalisasi.

ϵ : Konstanta kecil yang ditambahkan untuk mencegah pembagian dengan nol pada proses perhitungan laju perubahan *loss*.

β : Parameter sensitivitas pembobotan.

Dengan demikian, selama proses *training* model akan secara otomatis memberikan perhatian lebih besar pada komponen *loss* yang masih sulit dioptimalkan, dan mengurangi perhatian pada komponen yang sudah mengalami perbaikan. Pendekatan ini memungkinkan proses optimasi menjadi lebih adaptif dibandingkan penggunaan bobot tetap, karena dalam praktiknya, kontribusi setiap *loss* tidak selalu perlu seimbang karena kebutuhan optimasi dapat berbeda pada setiap fase *training* (Heydari dkk., 2019). Meskipun rasio optimal dapat dicari melalui *grid search*, pendekatan tersebut memerlukan proses pelatihan berulang yang mahal secara komputasi. Selain itu, rasio kontribusi *loss* juga dapat berubah selama proses *training* (Groenendijk dkk., 2020). Oleh karena itu, *adaptive loss*

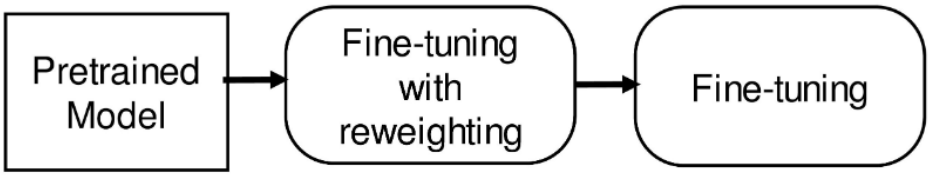
weighting digunakan untuk menyesuaikan bobot setiap *loss* secara dinamis berdasarkan kondisi pembelajaran, sehingga proses optimasi menjadi lebih fleksibel dan mampu memanfaatkan potensi *fusion loss* secara lebih efektif dibandingkan dengan bobot statis.

Pada skema *re-weighting loss*, mekanisme *warm-up* penting diterapkan karena penerapan *loss* yang kompleks terlalu awal berisiko mendistorsi representasi yang sedang dibentuk model sebelum pola dasar data dipelajari secara memadai. (Cao dkk., 2019) menunjukkan bahwa menunda penerapan *loss* yang kompleks hingga model membangun representasi awal yang stabil terbukti efektif untuk menghindari komplikasi optimasi di awal pelatihan pada kondisi data tidak seimbang.

Penelitian (Valizadehaslani dkk., 2022) mengusulkan strategi *two-stage fine-tuning* untuk menangani *class imbalance* pada data yang tidak seimbang. Pendekatan ini didasarkan pada temuan bahwa efek *reweighting* paling efektif di awal pelatihan dan kelas minoritas cenderung mengalami *representational collapse* pada layer atas jika tidak mendapat perlakuan khusus sejak awal. Penelitian ini juga sejalan dengan rekomendasi dari penelitian (Sushma dkk., 2025) yang menyatakan bahwa *fine-tuning* pada model *pretrained* seperti BERT dan XLM-RoBERTa merupakan arah penelitian lanjutan penelitian tersebut.

Eksperimen pada dataset SST-2 dengan rasio *imbalance* 0,2 menunjukkan bahwa *two-stage fine-tuning* meningkatkan *f1-score* kelas minoritas sebesar 1,6% dibandingkan *vanilla fine-tuning.* Selain itu, metode ini juga menunjukkan kemampuan generalisasi lebih baik pada data *out-of-distribution* (OOD) karena

fitur *pretrained* yang dipertahankan di tahap pertama membantu model beradaptasi pada distribusi data baru. *Flowchart* untuk *two-stage fine-tuning* dapat dilihat pada Gambar 2.5.



Gambar 2. 5 *Two-stage fine-tuning flowchart* (Valizadehaslani dkk., 2022)

Beberapa penelitian menunjukkan bahwa *imbalance data handling* juga dapat dilakukan pada tahap inferensi melalui *threshold tuning* untuk mengoptimalkan batas keputusan guna meningkatkan performa kelas minoritas tanpa perlu mengubah distribusi data latih maupun melatih ulang model (Esposito dkk., 2021).

2.1.10. Comparative Study

Comparative Study merupakan pendekatan penelitian yang dilakukan dengan membandingkan dua atau lebih metode, model ataupun konfigurasi secara sistematis untuk mengidentifikasi perbedaan performa, karakteristik, maupun efektivitas dari metode yang diuji (Zohreh Dehdashti Shahrokh & Seyed Mojtaba Miri, 2019). Dalam bidang *machine learning, comparative study* umum digunakan untuk mengevaluasi beberapa pendekatan pada dataset dan skenario eksperimen yang sama sehingga hasil perbandingan dapat dilakukan secara adil. Pendekatan ini juga membantu peneliti menentukan metode dengan performa terbaik berdasarkan metrik evaluasi tertentu seperti *accuracy, precision, recall* dan *f1-score* (Bogatinovski dkk., 2021).

Perbedaanya dengan *ablation study* adalah tujuannya, karena *ablation study* memiliki tujuan yang berbeda, yaitu menganalisis kontribusi suatu komponen terhadap performa metode dengan cara menghapus, memodifikasi atau menonaktifkan komponen tertentu dari satu arsitektur utama (Sheikholeslami dkk., 2025).

2.1.11. Evaluation Metrics

Model *machine learning* maupun *deep learning* perlu dievaluasi karena performa tinggi pada data pelatihan belum tentu menunjukkan kemampuan model pada data baru, evaluasi diperlukan untuk memastikan model benar-benar bekerja dengan baik dan tidak bias terhadap distribusi data tertentu (Dr. Sheshang Degadwala & Dhairya Vyas, 2024).

1. Accuracy

Accuracy merupakan metrik yang mengukur proporsi prediksi yang benar terhadap seluruh data yang dievaluasi, dengan persamaan 8.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Metrik ini memberikan gambaran umum performa model, namun kurang representatif pada data dengan distribusi kelas tidak seimbang karena dapat dipengaruhi oleh dominasi kelas minoritas.

2. Precision

Precision mengukur tingkat ketepatan prediksi positif yang dihasilkan model dengan persamaan 9.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Nilai *precision* yang tinggi menunjukkan bahwa sebagian besar data yang diprediksi sebagai positif memang benar-benar positif, sehingga meminimalkan kesalahan *False Positive.*

3. Recall

Recall mengukur kemampuan model dalam mendeteksi seluruh data positif yang sebenarnya dengan persamaan 10.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Nilai *recall* yang tinggi menunjukkan bahwa model mampu meminimalkan kesalahan *False Negative* dan mendeteksi mayoritas *instance* positif.

4. F1-Score

F1-Score merupakan rata-rata harmonik antara *precision* dan *recall,* dengan persamaan 11.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

F1-Score digunakan untuk memberikan evaluasi yang seimbang antara *precision* dan *recall,* terutama pada kasus klasifikasi dengan distribusi data yang tidak seimbang.

Keterangan:

True Positive (TP)	: Jumlah data yang diprediksi positif dan berlabel positif pada data sebenarnya.
True Negative (TN)	: Jumlah data yang prediksi negatif dan berlabel negatif pada data sebenarnya.
False Positive (FP)	: Jumlah data yang di prediksi positif, tetapi berlabel negatif pada data sebenarnya.

False Negative (FN) : Jumlah data yang di prediksi negatif, tetapi berlabel positif pada data sebenarnya.

Penelitian ini mengutamakan *f1-score* sebagai indikator utama evaluasi model, khususnya *f1-score* pada kelas minoritas, karena distribusi data latih yang tidak seimbang. Pada kondisi tersebut, akurasi dapat memberikan hasil yang bias karena model dapat memperoleh nilai tinggi dengan lebih banyak memprediksi kelas mayoritas (*accuracy paradox/bias accuracy*). Sebaliknya, *f1-score* menggabungkan *precision* dan *recall* melalui rata-rata harmonis, sehingga mampu mengevaluasi keseimbangan antara ketepatan prediksi kelas positif dan kemampuan model dalam mendeteksi sampel kelas minoritas. Oleh karena itu, penggunaan *f1-score* kelas minoritas memberikan evaluasi yang lebih representatif terhadap kemampuan model dalam menangani ketidakseimbangan kelas. (Abdelhamid & Desai, 2024).

2.1.12. Confusion Matrix

Confusion Matrix atau matriks kebingungan adalah sebuah tabel tabulasi silang yang merepresentasikan perbandingan antara hasil prediksi model dan kelas aktual data (Sokolova & Lapalme, 2009).

Data class	Classified as *pos*	Classified as *neg*
pos	true positive (*tp*)	false negative (*fn*)
neg	false positive (*fp*)	true negative (*tn*)

$$\begin{bmatrix} tp & fn \\ fp & tn \end{bmatrix}$$

Gambar 2. 6 *Confusion Matrix* (Sokolova & Lapalme, 2009)

Dalam konteks deteksi komentar *toxic,* matriks ini memetakan empat fundamental, yaitu *True Positive* (TP) ketika komentar *toxic* diprediksi benar sebagai *toxic, True Negative* (TN) ketika komentar netral diprediksi benar sebagai

netral, *False Positive* (FP) ketika komentar netral keliru diprediksi sebagai *toxic, False Negative* (FN) ketika komentar *toxic* gagal dideteksi dan dianggap netral.

2.2. State-of-the-art

Tabel 2. 2 *State-of-the-art* penelitian

Penulis	**Judul**	**Dataset**	**Metode**	**Temuan**	**Batasan**
(Sulistiyawati dkk., 2022)	Prediksi Kata Kasar Berbahasa Indonesia Menggunakan *Machine Learning* Berbasis *Mobile Infrastructure*	Dataset teks berbahasa Indonesia terkait kata kasar yang dikompilasi dari beberapa penelitian sebelumnya.	Pendekatan *machine learning* klasik menggunakan TF-IDF sebagai ekstraksi fitur dan Naïve Bayes sebagai *classifier*	Model Naïve Bayes mencapai akurasi 84.26%, *recall* 86.81%, dan precision 83.15% pada data uji, menunjukkan bahwa pendekatan klasik masih dapat digunakan untuk deteksi kata kasar bahasa Indonesia dalam skenario aplikasi nyata	Mengalami *overfitting*, hanya menangani bahasa Indonesia *monolingual*, tidak mempertimbangkan konteks semantik maupun *cross-lingual*, serta belum menggunakan *deep learning.*
(Salsabila Azzahra dkk., 2021)	*Toxic Comment Classification on Social Media Using Support Vector Machine*	*7k multi-label Indo Toxic Social Media Comment.*	Kombinasi TF-IDF, Chi-Square *feature selection* dan SVM.	Model SVM dengan kernel linear mencapai performa terbaik dengan f1-score 76,57% menunjukkan metode	Representasi berbasis kata bersifat dangkal sehingga sering gagal menangkap konteks, performa relatif lebih rendah dibandingkan

Penulis	Judul	Dataset	Metode	Temuan	Batasan
	and Chi Square Feature Selection			tradisional masih kompetitif dengan dataset terbatas	dengan model *deep learning,* tidak adaptif terhadap variasi bahasa.
(Cheng, 2024)	Dataset *Augmentation for Counteracting Bias in Toxic Comment Classification*	Jigsaw *Toxic Comment* Dataset.	Teknik *Data Augmentation* dengan pendekatan TF-IDF dan *Machine Learning* klasik.	Augmentasi data membantu meningkatkan *recall* dan *f1-score,* serta mengurangi *bias* prediksi terhadap kelompok tertentu.	Belum menggunakan Transformer serta tidak membahas skenario *cross lingual.*
(Made dkk., 2025)	*Browser-Based Detection of Harmful Content with Deep Learning Model*	Kombinasi *Multi-label* 13k kaggle dataset dan 2,8k data *tweet* dari Twitter.	Integrasi BiLSTM sebagai *embedding* dengan Whisper ASR.	BiLSTM dengan FastText menghasilkan performa tinggi dan sistem berhasil diterapkan dalam aplikasi nyata berbasis *browser extension.*	Objek penelitian masih bersifat *monolingual* dan belum mengeksplorasi pendekatan *cross-lingual* atau *transfer* pengetahuan antar bahasa.

Penulis	**Judul**	**Dataset**	**Metode**	**Temuan**	**Batasan**
(Neog & Baruah, 2025)	*A Multi-Layer Hybrid Deep Learning Model for Toxic Comment Detection in Assamese*	*Custom-built Assamese social media* dataset	Model *deep learning hybrid multi-layer* yang mengombinasikan CNN, BiLSTM, dan BiGRU dengan *embedding* FastText serta *hyperparameter tuning* menyeluruh.	*Model* CNN-BiLSTM-BiGRU memberikan performa terbaik dengan akurasi 94,25% dan *f1-score* 0,93%.	Hanya fokus pada satu bahasa (Assam), dan tidak mendukung skenario *cross-lingual.*
(Nabiilah dkk., 2024)	*Indonesian multilabel classification using IndoBERT embedding and MBERT classification*	*7k multi-label Indo Toxic Social Media Comment.*	IndoBERT sebagai *feature extraction* dan mBERT sebagai *classifier.*	Kombinasi dua *pre-trained model* BERT mampu meningkatkan performa klasifikasi *toxic comment* bahasa Indonesia dengan *f1-score* mencapai 90% menunjukkan efektivitas representasi kontekstual	Masih berfokus pada skenario *monolingual* bahasa Indonesia, belum melakukan eskplorasi evaluasi pada data *cross-lingual.*

Penulis	Judul	Dataset	Metode	Temuan	Batasan
				transformer pada data media sosial.	
(Shahid dkk., 2024)	*Leveraging deep learning for toxic comment detection in cursive languages*	Dataset *toxic* bahasa Urdu.	*Fine-tuning* BERT, perbandingan dengan GPT-2.	BERT-*based model* mengugguli CNN dan LSTM. *Fine-tuning* transformer menghasilkan *f1-score* tinggi yaitu 85% meskipun dataset terbatas, menunjukkan efektivitas *pretrained model* pada bahasa *low resource.*	Studi *monolingual,* tidak mengeksplor skenario *cross-lingual.*
(Hatvani & Yang, 2025)	*Automated detection of toxic comments in Hungarian*	655 komentar Hungarian.	huBERT, mBERT, SetFit.	Transformer menunjukkan performa lebih baik dibandingkan dengan model klasik pada kondisi *low data resource.*	Dataset sangat kecil, fokus pada *monolingual* dan tidak mengevaluasi skenario *cross-lingual.*
(Sushma dkk., 2025)	*Enhanced toxic comment*	Jigsaw *Toxic*	Kombinasi *Long Short Term Memory*	Kombinasi LSTM dengan BERT *embeddings* yang	BERT pada penelitian tersebut hanya digunakan

Penulis	Judul	Dataset	Metode	Temuan	Batasan
	detection model through Deep Learning models using Word embeddings and transformer *architectures*	*Comment* Dataset	(LSTM) sebagai *classifier* dan *Bidirectional Encoder from Transformer* (BERT) sebagai *Feature Extractor.*	mencapai akurasi 92.11%, menunjukkan bahwa *contextual embeddings* berbasis transformer meningkatkan performa deteksi komentar *toxic.*	sebagai *Feature Extraction* sehingga belum dioptimalkan secara langsung terhadap karakteristik toksisitas *(fine-tuning)*. Selain itu, penelitian tersebut masih terbatas pada bahasa Inggris dan belum mengeksplorasi skenario *cross-lingual*, sehingga membuka peluang penelitian menggunakan model *cross-lingual* seperti XLM-RoBERTa.
(Lashkarashvili & Tsintsadze, 2022)	*Toxicity detection in*	10k komentar Georgian.	CNN, RNN, BiLSTM dan Transformer.	CNN mengugguli RNN/LSTM maupun Transformer. CNN	Fokus *monolingual,* tidak mengevaluasi skenario *cross-lingual.*

Penulis	Judul	Dataset	Metode	Temuan	Batasan
	online Georgian discussions			mencapai akurasi mencapai akurasi 87%.	
(Grotti, 2024)	*Italian Linguistic Features for Toxic Language Detection in Social Media*	HaSpeeDe, *Italian Toxic Dataset*	*Fine-tuning* Italian-BERT.	*Fine-tuned* BERT memberikan performa tinggi dengan akurasi 88%.	Terbatas pada bahasa Italia dan tidak melakukan eksplorasi skenario *cross lingual.*
(Song, Huang, & Zhang, 2021)	*A hybrid model for monolingual and multilingual toxic comment detection*	Jigsaw *Multilingual Toxic Comment* Dataset	*Ensemble Model* yang menggabungkan BERT *monolingual* dan XLM-RoBERTa *multilingual.*	*Ensemble model* mengungguli *fine-tuning* XLM-R standar, menunjukkan bahwa kombinasi akurasi tinggi model *monolingual* dan generalisasi model *multilingual* dapat meningkatkan performa deteksi *toxic* multibahasa.	Membutuhkan model *monolingual* terpisah untuk setiap bahasa, arsitektur lebih kompleks.

Penulis	**Judul**	**Dataset**	**Metode**	**Temuan**	**Batasan**
(Song, Huang, & Xiao, 2021)	*A study of multilingual toxic text detection approaches under imbalanced sample distribution*	Jigsaw *Multilingual Toxic Comment* Dataset	mBERT dan XLM-RoBERTa.	XLM-RoBERTa mengugguli performa mBERT pada kondisi data tidak seimbang.	Memanfaatkan fusi *loss* untuk mengatasi ketidakseimbangan data tapi masih menggunakan *static loss weighting* dan belum melakukan eksplorasi terhadap *fine-tuning model* dan *adaptive loss weighting* serta masih memanfaatkan data hasil augmentasi translasi.
(El-Alami dkk., 2022)	*A multilingual offensive language detection method based on transfer learning from*	*Bilingual English-Arabic* Dataset	*Fine-tuned BERT-based multilingual models.*	Hasil penelitian menunjukkan bahwa *fine-tuning* model transformer berbasis *multilingual* meningkatkan performa deteksi bahasa ofensif	Data target dihasilkan dengan cara augmentasi translasi dan belum menggunakan dataset asli bahasa target.

Penulis	Judul	Dataset	Metode	Temuan	Batasan
	transformer *fine-tuning model*			pada skenario *multilingual*.	
(Mozafari dkk., 2022)	*Cross-Lingual Few-Shot Hate Speech and Offensive Language Detection Using Meta Learning*	Gabungan 15 dataset *hate speech* (8 bahasa) dan 6 dataset *offensive language* (6 bahasa) dari media sosial (Twitter).	Pendekatan *meta learning* (MAML & Proto-MAML) dengan XLM-RoBERTa sebagai *base model.*	Proto-MAML dengan XLM-R secara konsisten mengungguli *fine-tuning* XLM-R standar pada skenario *few-shot cross-lingual*.	Kompleksitas pelatihan tinggi, memerlukan konstruksi *episodic task.*
(Mengxin & Shaolin, 2025)	*UD-KD: A Structure-Aware Framework for Zero-Shot Cross-Lingual*	Dataset komentar berbahasa Inggris (Wikipedia	Kerangka UD-KD *(Unsupervised Distillation - Knowledge Distillation)* berbasis	Pendekatan UD-KD mampu meningkatkan performa deteksi *offensive language* lintas bahasa dibanding *baseline zero-*	Membutuhkan LLM berskala besar dan sumber daya komputasi tinggi, tidak mengeksplor *fine-tuning* model multibahasa

Penulis	Judul	Dataset	Metode	Temuan	Batasan
	Offensive Language Detection of Large Language Models	*Toxic Comments,* sekitar 160k data) sebagai bahasa sumber, dengan evaluasi *zero-shot* pada bahasa target.	*Large Language Models* (LLaMA-2-7B dan SeaLLM-7B) untuk mentransfer pengetahuan ke model yang lebih kecil secara *zero-shot cross-lingual*, tanpa data berlabel bahasa target.	*shot*, menunjukkan bahwa LLM dapat mentransfer pengetahuan lintas bahasa tanpa *fine-tuning* pada bahasa target.	ringan (mBERT/XLM-R), sehingga kurang praktis untuk skenario dengan keterbatasan komputasi.
(Bell dkk., 2025)	*Translate, then Detect: Leveraging Machine Translation for Cross-Lingual*	10 *benchmark* dataset toksisitas dari 17 bahasa,	Pendekatan *translate-classify*, yaitu menerjemahkan teks non-Inggris ke bahasa Inggris menggunakan sistem	Pendekatan *translate-classify* secara konsisten mengungguli *classifier multilingual out-of-distribution* pada 81,3% bahasa, dan sering kali	Sangat bergantung pada kualitas sistem terjemahan, Jika hasil terjemahan kurang baik, performa deteksi toksisitas juga ikut menurun, terutama pada

Penulis	Judul	Dataset	Metode	Temuan	Batasan
	Toxicity Classification	tidak termasuk Bahasa Indonesia.	*Machine Translation* kemudian melakukan klasifikasi toksisitas dengan *classifier* berbahasa Inggris.	menyaingi bahkan melampaui *classifier* khusus bahasa, manfaat terjemahan berkorelasi kuat dengan kualitas MT dan tingkat *resource* bahasa, serta *classifier* tradisional lebih stabil dibanding LLM *judge* pada bahasa *low-resource.*	bahasa dengan sumber daya terbatas. Selain itu, metode ini tidak melakukan *fine-tuning* pada data hasil terjemahan, sehingga kemampuan adaptasi model menjadi terbatas.

Secara garis besar, kajian deteksi *toxic comment* menunjukkan perkembangan yang cukup konsisten dari waktu ke waktu. Penelitian awal banyak memanfaatkan metode tradisional seperti TF-IDF, Naïve Bayes dan SVM yang relatif sederhana dan masih efektif pada skenario tertentu. Namun, pendekatan ini kerap mengalami keterbatasan serius, terutama dalam menangkap konteks semantik dan lintas bahasa. Seiring meningkatnya kebutuhan tersebut, penelitian kemudian bergeser ke metode *deep learning* berbasis CNN dan RNN yang menawarkan representasi fitur lebih kaya, meskipun sebagian besar masih berfokus pada satu bahasa dan belum

di rancang untuk menangani perbedaan lintas bahasa. Perkembangan selanjutnya ditandai dengan pemanfaatan model transformer seperti BERT, IndoBERT dan mBERT yang mampu meningkatkan performa melalui mekanisme *self-attention* nya. Meski demikian, sebagian besar studi masih terbatas pada skenario *monolingual*. Penelitian-penelitian terbaru menunjukkan bahwa XLM-RoBERTa memiliki potensi yang lebih kuat dalam menangani perbedaan bahasa dibandingkan dengan mBERT, bahkan pada kondisi data tidak seimbang, namun pemanfaatannya masih dominan pada evaluasi *multilingual* bahasa lain dan memiliki celah dibagian penanganan data tidak seimbang. Hingga saat ini belum ditemukan penelitian yang secara khusus mengkaji deteksi *cross-lingual toxic comment* bahasa Indonesia dan bahasa Inggris. Kekosongan kajian ilmiah inilah yang menjadi salah satu *research gap* yang ditemukan.

2.3. Matriks Penelitian

Tabel 2. 3 Matriks Penelitian

No	Peneliti	Skenario Bahasa				Algoritma											Distribusi Data		Penanganan *Imbalance*	
		Monolingual	*Bilingual*	*Multilingual*	*Cross-lingual*	BERT-based	MBERT	XLM-R	CNN	BiLSTM	RNN/GRU	SVM	NB	Meta	KD	MT	Seimbang	Tidak Seimbang	Ya	Tidak
1	(Sulistiyawati dkk., 2022)	✓											✓					✓		✓
2	(Salsabila Azzahra dkk., 2021)	✓										✓						✓		✓
3	(Cheng, 2024)	✓																✓		✓

No	Peneliti	Skenario Bahasa				Algoritma											Distribusi Data		Penanganan *Imbalance*	
		Monolingual	*Bilingual*	*Multilingual*	*Cross-lingual*	BERT-based	MBERT	XLM-R	CNN	BiLSTM	RNN/GRU	SVM	NB	Meta	KD	MT	Seimbang	Tidak Seimbang	Ya	Tidak
4	(Made dkk., 2025)	✓								✓								✓		✓
5	(Neog & Baruah, 2025)	✓							✓	✓	✓						✓		-	-
6	(Nabiilah dkk., 2024)	✓				✓	✓											✓		✓
7	(Shahid dkk., 2024)	✓				✓												✓		✓

No	Peneliti	Skenario Bahasa				Algoritma											Distribusi Data		Penanganan *Imbalance*	
		Monolingual	*Bilingual*	*Multilingual*	*Cross-lingual*	BERT-based	MBERT	XLM-R	CNN	BiLSTM	RNN/GRU	SVM	NB	Meta	KD	MT	Seimbang	Tidak Seimbang	Ya	Tidak
8	(Hatvani & Yang, 2025)	✓				✓	✓											✓		✓
9	(Sushma dkk., 2025)	✓				✓					✓							✓		✓
10	(Lashkarashvili & Tsintsadze, 2022)	✓							✓		✓						✓		-	-
11	(Grotti, 2024)	✓				✓												✓		✓

No	Peneliti	Skenario Bahasa				Algoritma										Distribusi Data		Penanganan *Imbalance*	
		Monolingual	*Bilingual*	*Multilingual*	*Cross-lingual*	BERT-based	MBERT	XLM-R	CNN	BiLSTM	RNN/GRU	SVM	NB	Meta	KD	MT	Seimbang	Tidak Seimbang	Ya	Tidak
12	(Song, Huang, & Zhang, 2021)	✓		✓	✓	✓		✓										✓		✓
13	(Song, Huang, & Xiao, 2021)			✓	✓		✓	✓										✓	✓	
14	(El-Alami dkk., 2022)			✓	✓	✓	✓									✓	✓		-	-
15	(Mozafari dkk., 2022)			✓	✓	✓		✓						✓				✓		✓

No	Peneliti	Skenario Bahasa				Algoritma											Distribusi Data		Penanganan *Imbalance*	
		Monolingual	*Bilingual*	*Multilingual*	*Cross-lingual*	BERT-based	MBERT	XLM-R	CNN	BiLSTM	RNN/GRU	SVM	NB	Meta	KD	MT	Seimbang	Tidak Seimbang	Ya	Tidak
16	(Mengxin & Shaolin, 2025)			✓	✓	✓									✓			✓		✓
17	(Bell dkk., 2025)			✓	✓											✓		✓		✓
18	Penelitian Ini		✓		✓			✓										✓	✓	