BAB 2
LANDASAN TEORI

2.1 Studi Literatur

Penelitian awal mengenai deteksi ujaran kebencian berbahasa Indonesia banyak memanfaatkan arsitektur *deep learning*. Imaduddin et al. mengimplementasikan model *Long Short-Term Memory* (LSTM) yang diintegrasikan dengan *word embedding* GloVe pada 13.169 *tweet* berbahasa Indonesia, menghasilkan *F1-score* sebesar 94% dan *recall* yang sangat tinggi mencapai 99% [21]. Meskipun efektif pada konteks monolingual, pendekatan ini masih terbatas dalam hal skalabilitas terhadap teks multibahasa. Pada konteks yang lebih luas, García et al. mengevaluasi berbagai model *Transformer*, termasuk ALBETO, TwHIN, DistilBETO, MarIA, dan BERT, pada dataset bahasa Spanyol dan Inggris dengan pendekatan *zero-shot* dan *few-shot learning* [20]. Hasil penelitian ini mendemonstrasikan bahwa *pre-trained language models* memiliki kemampuan adaptasi yang menakjubkan dalam menangani tugas deteksi ujaran kebencian lintas bahasa, bahkan dengan data pelatihan yang terbatas.

Fenomena *code-mixing* semakin mendapat perhatian dalam penelitian deteksi ujaran kebencian. Kathiravan et al. mengusulkan metode *Sentence Transfer Fine-tuning* (SetFit) yang dikombinasikan dengan regresi logistik untuk mendeteksi konten ofensif pada teks campuran Tamil-Inggris dan berhasil mengungguli model-model tradisional seperti mBERT, LSTM, dan BERT dengan akurasi sebesar 89,72% [37]. Pada konteks *code-mixing* Hindi-Inggris, Jha et al. mengembangkan pendekatan berbasis TabNet yang memanfaatkan fitur hasil ekstraksi dari MuRIL, dengan kombinasi TabNet + PCA + MuRIL mencapai akurasi tertinggi sebesar 90% [38]. Sementara itu, Pamungkas et al. melakukan studi awal mengenai deteksi ujaran kebencian pada media sosial Indonesia dengan fokus pada teks *code-mixing* yang melibatkan bahasa Jawa dan Sunda, dan mengungkapkan bahwa performa model cenderung menurun hingga 4–7% ketika menangani teks *code-mixing* dibandingkan teks monolingual [10]. Penelitian terkini oleh Wicaksana et al. membandingkan model BLOOM dan XLM-RoBERTa pada teks campuran bahasa Indonesia dan bahasa Inggris menggunakan dataset 30.000 sampel hasil augmentasi berbasis GPT, di mana BLOOM mencapai akurasi dan *F1-score* sebesar 99,97%, mengungguli XLM-RoBERTa yang mencapai 98,69% [17]. Namun demikian,

penelitian tersebut sepenuhnya bergantung pada data sintetis sehingga kemampuan model dalam menangani variasi bahasa informal nyata seperti kata *slang* atau bahasa gaul dan singkatan tidak baku belum dieksplorasi secara mendalam.

Mempertimbangkan kesenjangan-kesenjangan tersebut, pemilihan model dalam penelitian ini didasarkan pada beberapa pertimbangan empiris yang bersumber dari literatur yang telah ditinjau. Dinarta et al. menunjukkan bahwa XLM-RoBERTa mencapai *F1-score* kompetitif pada teks *code-mixing* Indonesia-Inggris, namun pendekatan tersebut mengandalkan arsitektur *encoder-only* yang dioptimalkan terutama untuk teks formal [19]. Sementara itu, Pamungkas et al. membuktikan bahwa teks *code-mixing* informal berbahasa Indonesia menghadirkan tantangan yang berbeda secara fundamental dari teks formal, sehingga membutuhkan model yang secara inheren dilatih pada variasi bahasa yang beragam [39]. Dalam perbandingan langsung antara kedua model, Wicaksana et al. melaporkan bahwa model BLOOM melampaui XLM-RoBERTa secara konsisten pada konteks *code-mixing* Indonesia-Inggris [17]. Lebih jauh, Winata et al. mendemonstrasikan bahwa model multibahasa berbasis *Transformer* secara umum lebih efektif dari model monolingual dalam menangani fenomena *code-mixing* [9]. Oleh karena itu, BLOOM-560M dipilih sebagai model dasar dalam penelitian ini karena kemampuannya memproses teks multibahasa informal secara terpadu, dengan *tokenizer Byte Pair Encoding* (BPE) yang secara alami menangani variasi *subword* seperti kata *slang* atau bahasa gaul dan singkatan tidak baku.

Berdasarkan tinjauan literatur di atas, terdapat beberapa *gap* penelitian yang signifikan. Pertama, sebagian besar penelitian hanya berfokus pada deteksi ujaran kebencian di tingkat kalimat tanpa disertai kemampuan untuk melokalisasi *hate-bearing* token yang bersifat ofensif, sehingga tidak ada *explainability* pada tingkat token. Kedua, dataset yang digunakan umumnya belum mencerminkan karakteristik bahasa informal media sosial secara komprehensif, khususnya yang mencakup kata *slang* atau bahasa gaul dan singkatan tidak baku seperti yang lazim ditemui pada percakapan daring berbahasa Indonesia. Ketiga, belum ada penelitian yang mengintegrasikan deteksi tingkat kalimat dan identifikasi *hate span* dalam satu sistem terpadu untuk teks *code-mixing* Indonesia-Inggris yang bersifat informal, khususnya dengan pendekatan pelabelan BIO otomatis berbasis leksikon tanpa ketergantungan pada sumber daya eksternal maupun anotasi manual. Tinjauan literatur ini dengan demikian memberikan landasan teoritis bagi pengembangan sistem yang diusulkan, sementara penelitian-penelitian lain yang turut berkontribusi pada bidang ini disajikan secara komprehensif pada Tabel 2.1.

Tabel 2.1. Analisis komparatif model NLP untuk deteksi ujaran kebencian

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
[10]	Hate Speech Detection on Indonesian Social Media: A Preliminary Study on Code-Mixed Language Issue	Linear SVM	Javanese	Social Media	84,9%	76,2%	78,9%	74,4%	Pamungkas et al. mengkaji deteksi ujaran kebencian pada teks campuran Jawa-Indonesia dan Sunda-Indonesia dengan membandingkan model ML tradisional (*Linear* SVM, *Logistic Regression*, *Random Forest*) dan DL (LSTM, GRU) pada data asli maupun hasil terjemahan Google Translate, dan menemukan bahwa model tradisional berfitur leksikal terbaik pada data Jawa-Indonesia, LSTM terbaik pada data Sunda-Indonesia, serta penerjemahan ke bahasa yang lebih kaya sumber daya justru menurunkan performa karena mengubah makna dan konteks ujaran kebencian dalam *tweet*.
		Linear SVM	Javanese-Indonesia	Social Media	79,8%	67,4%	70%	66%	
		Linear SVM	Javanese-English	Social Media	80,4%	67,3%	71,2%	65,5%	
		Linear SVM	Sundanese	Social Media	79,4%	70,8%	73%	69,5%	
		Linear SVM	Sundanese-Indonesia	Social Media	77%	67,9%	69,4%	67%	
		Linear SVM	Sundanese-English	Social Media	78,7%	69,4%	72%	68%	
		Decision Tree	Javanese	Social Media	83,9%	75,1%	76,8%	73,9%	
		Decision Tree	Javanese-Indonesia	Social Media	77,6%	63,4%	65,8%	62,2%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		Decision Tree	Javanese-English	Social Media	77,2%	62,9%	65,1%	61,8%	
		Decision Tree	Sundanese	Social Media	77,1%	68,1%	69,6%	67,2%	
		Decision Tree	Sundanese-Indonesia	Social Media	74,7%	65%	66%	64,4%	
		Decision Tree	Sundanese-English	Social Media	75%	67,8%	67,5%	68,2%	
		Logistic Regression	Javanese	Social Media	86,3%	76,9%	83%	73,8%	
		Logistic Regression	Javanese-Indonesia	Social Media	81,7%	67,5%	74,8%	65%	
		Logistic Regression	Javanese-English	Social Media	82%	65,4%	77,8%	63%	
		Logistic Regression	Sundanese	Social Media	81,1%	71,7%	76,6%	69,6%	
		Logistic Regression	Sundanese-Indonesia	Social Media	78,1%	65,6%	71,7%	64%	
		Logistic Regression	Sundanese-English	Social Media	79,2%	66,5%	74,6%	64,5%	
		Random Forest	Javanese	Social Media	84,1%	70,4%	82,6%	67%	
		Random Forest	Javanese-Indonesia	Social Media	80,6%	58,8%	77,6%	58,1%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		Random Forest	Javanese-English	Social Media	79%	59,2%	68,3%	58,3%	
		Random Forest	Sundanese	Social Media	80,1%	68%	76,6%	65,7%	
		Random Forest	Sundanese-Indonesia	Social Media	79,2%	65,1%	76%	63,2%	
		Random Forest	Sundanese-English	Social Media	78,7%	68,4%	72,3%	66,7%	
		LSTM	Javanese	Social Media	85,1%	73,9%	81,9%	70,6%	
		LSTM	Javanese-Indonesia	Social Media	80,8%	61,9%	74,8%	60,2%	
		LSTM	Javanese-English	Social Media	80,8%	66,8%	72,3%	64,7%	
		LSTM	Sundanese	Social Media	78,6%	72,7%	72,1%	73,3%	
		LSTM	Sundanese-Indonesia	Social Media	77,1%	69,8%	69,9%	69,7%	
		LSTM	Sundanese-English	Social Media	75,6%	69,5%	68,7%	70,5%	
		GRU	Javanese	Social Media	84,9%	73,2%	82,3%	69,7%	
		GRU	Javanese-Indonesia	Social Media	81,3%	62,8%	76,6%	60,9%	
		GRU	Javanese-English	Social Media	81,1%	64,1%	74,5%	62%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		GRU	Sundanese	Social Media	79,4%	72,2%	72,9%	71,6%	
		GRU	Sundanese-Indonesia	Social Media	75,9%	67,6%	68,1%	67,3%	
		GRU	Sundanese-English	Social Media	72,1%	67,3%	66,6%	70,2%	
[15]	Context-aware and Expert Data Resources for Brazilian Portuguese Hate Speech Detection	TF-IDF (SVM)	Portuguese	HateBR 2.0 corpus	–	84%	84%	84%	Vargas et al. mengembangkan HateBR 2.0, korpus 7.000 komentar Instagram politisi Brasil yang dianotasi ke dalam klasifikasi biner dan sembilan target *hate speech*, disertai MOL berisi 1.000 istilah pejoratif beranotasi konteks dalam enam bahasa, dengan *baseline* mBERT mencapai *F1-score* tertinggi 85%, melampaui *dataset hate speech* Portugis sebelumnya.
		TF-IDF (NB)	Portuguese	HateBR 2.0 corpus	–	77%	78%	78%	
		fastText-*unigram*	Portuguese	HateBR 2.0 corpus	–	84%	84%	84%	
		fastText-*bigram*	Portuguese	HateBR 2.0 corpus	–	81%	82%	81%	
		fastText-trigram	Portuguese	HateBR 2.0 corpus	–	80%	81%	80%	
		mBERT	Portuguese	HateBR 2.0 corpus	–	85%	85%	85%	
		MOL	English	HateBR 2.0 corpus	–	72%	74%	73%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		MOL	Spanish	HateBR 2.0 corpus	–	82%	82%	82%	
		MOL	Portuguese	HateBR 2.0 corpus	–	88%	88%	87%	
		B + M	English	HateBR 2.0 corpus	–	73%	74%	74%	
		B + M	Spanish	HateBR 2.0 corpus	–	79%	80%	80%	
		B + M	Portuguese	HateBR 2.0 corpus	–	83%	83%	83%	
[17]	Enhancing Hate Speech Detection in Mixed-Language Texts: A Comparative Study of BLOOM and XLM-RoBERTa Models	BLOOM	English-Indonesia	GPT Generated	99,97%	99,97%	100%	99,93%	Wicaksana et al. mengimplementasikan model BLOOM melalui *fine-tuning* dan penyetelan hiperparameter berbasis *grid search* pada dataset sintetis 30.000 sampel teks campuran Indonesia-Inggris yang dibangkitkan menggunakan GPT-4, sehingga berhasil mencapai akurasi 98,22% dan *F1-score* 98,23% pada data uji utama.
		XLM-RoBERTa	English-Indonesia	GPT Generated	98,96%	98,69%	98,71%	98,69%	
[21]	Application of LSTM and GloVe Word Embedding for Hate Speech Detection in Indonesian Twitter Data	GloVe + 6 Layer	Indonesia	Twitter	94,24%	94%	89%	99%	Imaduddin et al. mendeteksi *hate speech* pada data Twitter berbahasa Indonesia menggunakan kombinasi LSTM dan *word embedding* GloVe 200 dimensi pada 13.169 data yang diseimbangkan menjadi 15.796 data melalui kombinasi *up-sampling* dan *down-sampling*, dengan arsitektur terbaik mencapai *precision* 89%, *recall* 99%, *F1-score* 94%, dan *accuracy* 94,24%.

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		LSTM + 6 Layer	Indonesia	Twitter	93,61%	93%	88%	99%	
		GloVe + 5 Layer	Indonesia	Twitter	94,24%	93%	88%	99%	
		LSTM + 5 Layer	Indonesia	Twitter	93,8%	93%	89%	98%	
[20]	Leveraging Zero and Few-Shot Learning for Enhanced Model Generality in Hate Speech Detection in Spanish and English	ALBETO	Spanish	EXIST 2021-es	–	79,33%	78,45%	80,23%	García-Díaz et al. mengevaluasi generalisasi model generatif berbasis instruksi (Flan-T5, Flan-Alpaca, mT0, dan Llama-2) pada skema *zero-shot learning* dan *few-shot learning* untuk deteksi ujaran kebencian biner berbahasa Spanyol dan Inggris, dan menemukan bahwa *fine-tuning* tetap unggul dalam *F1-score*, namun Llama-2 13B paling kompetitif di antara model generatif dengan stabilitas lebih baik pada bahasa Inggris, sementara *few-shot learning* umumnya tidak mengungguli *zero-shot learning* akibat kualitas pemilihan contoh yang kurang representatif.
		ALBETO	Spanish	EXIST 2022-es	–	77,18%	79,05%	75,39%	
		ALBETO	Spanish	HatEval 2019	–	75,33%	70,25%	81,21%	
		ALBETO	Spanish	MisoCorpus 2020	–	89,33%	90,14%	88,54%	
		ALBETO	Spanish	Hate Football Corpus 2023	–	84,89%	85%	84,78%	
		ALBETO	Spanish	HaterNET 2019	–	59,15%	64,86%	54,37%	
		BETO	Spanish	EXIST 2021-es	–	78,96%	80,59%	77,4%	
		BETO	Spanish	EXIST 2022-es	–	80,57%	77,76%	83,6%	
		BETO	Spanish	HatEval 2019	–	73,44%	66,42%	82,12%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		BETO	Spanish	MisoCorpus 2020	–	89,92%	90,36%	89,49%	
		BETO	Spanish	Hate Football Corpus 2023	–	85,15%	85,26%	85,04%	
		BETO	Spanish	HaterNET 2019	–	67,59%	72,76%	63,11%	
		DistilBETO	Spanish	EXIST 2021-es	–	81,83%	80,38%	83,33%	
		DistilBETO	Spanish	EXIST 2022-es	–	80,76%	78,03%	83,68%	
		DistilBETO	Spanish	HatEval 2019	–	76,24%	70,58%	82,88%	
		DistilBETO	Spanish	MisoCorpus 2020	–	89,59%	89,96%	89,22%	
		DistilBETO	Spanish	Hate Football Corpus 2023	–	84,38%	83,72%	85,04%	
		DistilBETO	Spanish	HaterNET 2019	–	67,53%	67,42%	67,64%	
		DistilBETO	Spanish	HASOC 2021	–	86,57%	84,75%	88,47%	
		MarIA	Spanish	EXIST 2021-es	–	81,78%	80,55%	83,05%	
		MarIA	Spanish	EXIST 2022-es	–	80,71%	78,01%	83,6%	
		MarIA	Spanish	HatEval 2019	–	75,92%	71,28%	81,21%	
		MarIA	Spanish	MisoCorpus 2020	–	90,29%	89,86%	90,72%	
		MarIA	Spanish	Hate Football Corpus 2023	–	85,18%	87,53%	82,94%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		MarIA	Spanish	HaterNET 2019	–	66,45%	67,9%	65,05%	
		mBERT	Spanish	EXIST 2021-es	–	74,44%	73,22%	75,71%	
		mBERT	Spanish	EXIST 2022-es	–	78,15%	73,45%	83,51%	
		mBERT	Spanish	HatEval 2019	–	70,24%	65,69%	75,45%	
		mBERT	Spanish	MisoCorpus 2020	–	88,69%	89,12%	88,27%	
		mBERT	Spanish	Hate Football Corpus 2023	–	80,93%	84,14%	77,95%	
		mBERT	Spanish	HaterNET 2019	–	58,52%	68,39%	51,13%	
		mBERT	English	EXIST 2021-en	–	73,23%	70,08%	76,67%	
		mBERT	English	EXIST 2022-en	–	76,42%	73,87%	79,16%	
		mBERT	English	HASOC 2021	–	84,83%	80,74%	89,35%	
		mBERT	English	EDOS 2023	–	70,19%	72,84%	67,73%	
		mBERT	English	HatEval 2019	–	61,68%	45,39%	96,19%	
		mDeBERTa	Spanish	EXIST 2021-es	–	82,09%	81,97%	82,2%	
		mDeBERTa	Spanish	EXIST 2022-es	–	66,86%	50,24%	99,91%	
		mDeBERTa	Spanish	HatEval 2019	–	74,31%	67,24%	83,03%	
		mDeBERTa	Spanish	MisoCorpus 2020	–	90,49%	90,68%	90,31%	
		mDeBERTa	Spanish	Hate Football Corpus 2023	–	82,59%	80,05%	85,3%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		mDeBERTa	Spanish	HaterNET 2019	–	68,86%	66,67%	71,19%	
		mDeBERTa	English	EXIST 2021-en	–	64,89%	48,03%	100%	
		mDeBERTa	English	EXIST 2022-en	–	66,15%	49,47%	99,82%	
		mDeBERTa	English	HASOC 2021	–	85,68%	82,41%	89,22%	
		mDeBERTa	English	EDOS 2023	–	71,69%	75,14%	68,56%	
		mDeBERTa	English	HatEval 2019	–	62,59%	45,97%	98,09%	
		TwHIN	Spanish	EXIST 2021-es	–	82,26%	78,46%	86,44%	
		TwHIN	Spanish	EXIST 2022-es	–	57,48%	50,03%	67,54%	
		TwHIN	Spanish	HatEval 2019	–	74,56%	72,83%	76,36%	
		TwHIN	Spanish	MisoCorpus 2020	–	90,12%	90,62%	89,63%	
		TwHIN	Spanish	Hate Football Corpus 2023	–	83,97%	84,76%	83,2%	
		TwHIN	Spanish	HaterNET 2019	–	67,3%	66,04%	68,61%	
		TwHIN	English	EXIST 2021-en	–	77,28%	73,96%	80,91%	
		TwHIN	English	EXIST 2022-en	–	78,39%	75,69%	81,31%	
		TwHIN	English	HASOC 2021	–	86,76%	80,84%	93,61%	
		TwHIN	English	EDOS 2023	–	72,08%	72,84%	71,34%	
		TwHIN	English	HatEval 2019	–	63,98%	47,59%	97,54%	
		ALBERT	English	EXIST 2021-en	–	75,8%	73,03%	78,79%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		ALBERT	English	EXIST 2022-en	–	77,83%	73,97%	82,11%	
		ALBERT	English	HASOC 2021	–	85,19%	81,09%	89,72%	
		ALBERT	English	EDOS 2023	–	70,05%	74,39%	66,19%	
		ALBERT	English	HatEval 2019	–	59,51%	42,79%	97,62%	
		BERT	English	EXIST 2021-en	–	79,77%	76,24%	83,64%	
		BERT	English	EXIST 2022-en	–	79,68%	74,8%	85,24%	
		BERT	English	HASOC 2021	–	86,07%	82,6%	89,85%	
		BERT	English	EDOS 2023	–	73,79%	71,92%	75,77%	
		BERT	English	HatEval 2019	–	63,45%	47,02%	97,54%	
		DistilBERT	English	EXIST 2021-en	–	78,09%	74,45%	82,12%	
		DistilBERT	English	EXIST 2022-en	–	77,51%	74,65%	80,59%	
		DistilBERT	English	EDOS 2023	–	72,16%	77,2%	67,73%	
		DistilBERT	English	HatEval 2019	–	62,24%	45,63%	97,86%	
		RoBERTa	English	EXIST 2021-en	–	77,25%	71,99%	83,33%	
		RoBERTa	English	EXIST 2022-en	–	79,19%	74,7%	84,26%	
		RoBERTa	English	HASOC 2021	–	86,11%	83,43%	88,97%	
		RoBERTa	English	EDOS 2023	–	71,68%	74,25%	69,28%	
		RoBERTa	English	HatEval 2019	–	62,47%	46,18%	96,51%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
[37]	SetFit: A Robust Approach for Offensive Content Detection in Tamil-English Code-Mixed Conversations Using Sentence Transfer Fine-tuning	SetFit	Tamil-English	Sosial Media	89%	88%	90%	87%	Pannerselvam et al. mengusulkan pendekatan SetFit+*Logistic Regression* untuk mendeteksi konten ofensif pada teks *code-mixed* Tamil-Inggris menggunakan 35.139 komentar YouTube yang dianotasi ke dalam enam kelas dengan rasio pembagian data 70:30, sehingga menghasilkan akurasi sebesar 89,72%, presisi 0,90, *recall* 0,87, dan *F1-score* 0,88.
		mBERT	Tamil-English	Sosial Media	86%	84%	88%	84%	
		LSTM	Tamil-English	Sosial Media	76%	67%	70%	65%	
		BERT	Tamil-English	Sosial Media	78%	70%	72%	68%	
		indicBERT	Tamil-English	Sosial Media	80%	72%	74%	70%	
		LaBSE	Tamil-English	Sosial Media	84%	79%	80%	78%	
[38]	A Framework for Online Hate Speech Detection on Code-mixed Hindi-English Text and Hindi Text in Devanagari	TabNet + PCA + M-BERT	Hindi-English	Twitter	65%	65%	64%	66%	Chopra et al. mengusulkan deteksi ujaran kebencian Hindi-Inggris (transliterasi dan Devanagari) menggunakan *embedding* MuRIL yang direduksi via PCA dan diklasifikasikan dengan TabNet, mencapai akurasi 90% dan *F1-score* 89–90% (konsisten pada data Devanagari, akurasi 85%), serta mengungguli *baseline* fastText+SVM-RBF, word2vec+BiLSTM, dan mBERT+TF-IDF+DNN.
		TabNet + PCA + Indic-BERT	Hindi-English	Twitter	71%	71%	72%	71%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		TabNet + PCA + XLM-RoBERTa	Hindi-English	Twitter	84%	85%	84%	84%	
		TabNet + PCA + MuRIL	Hindi-English	Twitter	90%	89%	89%	89%	
[40]	Multi-class Hate Speech Detection in the Norwegian Language using FAST-RNN and Multilingual Fine-tuned *Transformer*s	mBERT	Norwegia	Resett, Twitter, Facebook	82%	79%	78%	82%	Hashmi et al. mengusulkan model FAST-RNN untuk deteksi ujaran kebencian multi-kelas pada bahasa Norwegia yang secara konsisten mengungguli berbagai model *Transformer* multibahasa maupun khusus Norwegia (mBERT, ELECTRA, FLAN-T5, Nor-BERT, Nor-T5, Scandi-BERT, nb-BERT) dalam hasil *hyperparameter tuning* dan *prompt-based fine-tuning*, dengan *precision*, *recall*, *accuracy*, dan *F1-score* sebesar 0,98, serta divalidasi lebih lanjut menggunakan LIME.
		ELECTRAbase	Norwegia	Resett, Twitter, Facebook	83%	75%	69%	83%	
		ELECTRAlarge	Norwegia	Resett, Twitter, Facebook	83%	75%	69%	83%	
		scandi-BERT	Norwegia	Resett, Twitter, Facebook	81%	81%	79%	81%	
		nb-BERTbase	Norwegia	Resett, Twitter, Facebook	81%	81%	79%	81%	
		nb-BERTlarge	Norwegia	Resett, Twitter, Facebook	81%	81%	81%	81%	
		Nor-BERTsmall	Norwegia	Resett, Twitter, Facebook	82%	79%	77%	83%	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		Nor-BERTbase	Norwegia	Resett, Twitter, Facebook	83%	80%	78%	83%	
		Nor-BERTlarge	Norwegia	Resett, Twitter, Facebook	83%	81%	80%	82%	
		FLAN-T5-small	Norwegia	Resett, Twitter, Facebook	80%	77%	76%	80%	
		FLAN-T5-base	Norwegia	Resett, Twitter, Facebook	83%	80%	82%	82%	
		Nor-T5small	Norwegia	Resett, Twitter, Facebook	77%	78%	77%	78%	
[41]	Fine-Grained Multilingual Hate Speech Detection Using Explainable AI and *Transformers*	mBERT	English	Public Dataset	97%	97%	–	–	Siddiqui et al. mengembangkan model deteksi ujaran kebencian berbutir halus multibahasa (Inggris, Urdu, dan Sindhi) pada lima kategori (*Disability*, *Gender/Sexual*, *Origin/Nationality*, *Race/Ethnicity*, *Religion*) menggunakan tiga model berbasis BERT yang diintegrasikan dengan LIME, dengan XLM-RoBERTa sebagai model terbaik mencapai *F1-score* 91% pada klasifikasi biner dan *F1-score* tertimbang 86% pada klasifikasi berbutir halus, dengan kategori *Religion* mencapai *F1-score* tertinggi sebesar 92%.
		mBERT	Urdu	Translated	69%	68%	–	–	
		mBERT	Sindhi	Translated from English	87%	86%	–	–	
		mBERT	Combined	Combined All Dataset	84%	84%	–	–	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		XLM-RoBERTa	English	Public Dataset	97%	97%	–	–	
		XLM-RoBERTa	Urdu	Translated	69%	68%	–	–	
		XLM-RoBERTa	Sindhi	Translated from English	92%	92%	–	–	
		XLM-RoBERTa	Combined	Combined All Dataset	85%	86%	–	–	
		Distil-RoBERTa	English	Public Dataset	97%	97%	–	–	
		Distil-RoBERTa	Urdu	Translated	54%	51%	–	–	
		Distil-RoBERTa	Sindhi	Translated from English	74%	73%	–	–	
		Distil-RoBERTa	Combined	Combined All Dataset	75%	74%	–	–	
[42]	HS-BAN: A Benchmark Dataset of Social Media Comments for Hate Speech Detection in Bangla	Bi-LSTM + FT(SG)	Bangla	Facebook, YouTube	–	86,85%	–	–	Romim et al. memperkenalkan HS-BAN, dataset *hate speech* biner Bangla terbesar (50.314 komentar Facebook/YouTube, tujuh kategori), dengan model Bi-LSTM + *embedding* FastText *skip-gram* informal mencapai *F1-score* tertinggi 86,85%, mengungguli *embedding* formal dan varian BERT.
		Bangla-BERT	Bangla	Facebook, YouTube	–	83,68%	–	–	
		CNN + BFT	Bangla	Facebook, YouTube	–	71,58%	–	–	
		mBERT-cased	Bangla	Facebook, YouTube	–	81,66%	–	–	

Lanjut pada halaman berikutnya

Tabel 2.1Analisis komparatif model NLP untuk deteksi ujaran kebencian (lanjutan)

Referensi	Judul	Model	Bahasa	Dataset	Accuracy	F1-score	Precision	Recall	State Of The Art (SOTA)
		mBERT-uncased	Bangla	Facebook, YouTube	–	82,56%	–	–	
		Bi-LSTM + BFT	Bangla	Facebook, YouTube	–	76,09%	–	–	
		CNN + FastText	Bangla	Facebook, YouTube	–	84%	–	–	
		Bi-LSTM + FastText	Bangla	Facebook, YouTube	–	83,89%	–	–	
		CNN + W2V(CBOW)	Bangla	Facebook, YouTube	–	80,03%	–	–	
		Bi-LSTM + W2V(CBOW)	Bangla	Facebook, YouTube	–	80,48%	–	–	
		CNN + W2V(SG)	Bangla	Facebook, YouTube	–	81,07%	–	–	
		Bi-LSTM + W2V(SG)	Bangla	Facebook, YouTube	–	80,93%	–	–	
		CNN + FT(CBOW)	Bangla	Facebook, YouTube	–	85,05%	–	–	
		Bi-LSTM + FT(CBOW)	Bangla	Facebook, YouTube	–	86,73%	–	–	
		CNN + FT(SG)	Bangla	Facebook, YouTube	–	86,85%	–	–	

2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan cabang ilmu komputer yang berada di persimpangan antara kecerdasan buatan dan linguistik komputasional, yang berfokus pada pengembangan sistem komputasional untuk memungkinkan mesin memahami, menginterpretasikan, dan menghasilkan bahasa manusia secara alami [43]. Tujuan utama NLP adalah menjembatani kesenjangan antara cara manusia berkomunikasi secara alami dan cara mesin memproses informasi, yang pada praktiknya menghadirkan tantangan kompleks karena bahasa manusia bersifat ambigu, kontekstual, dan terus berkembang [44].

Ruang lingkup NLP mencakup berbagai tugas komputasional yang tersebar pada beberapa tingkat abstraksi linguistik, mulai dari tokenisasi, *part-of-speech tagging*, dan *parsing* pada level sintaktis, hingga analisis sentimen, *named entity recognition*, serta deteksi ujaran kebencian pada tingkat semantik dan pragmatik [45]. Perkembangan NLP modern didorong oleh kemajuan arsitektur *deep learning*, khususnya model *Transformer* berbasis *pre-training* seperti BLOOM, BERT, dan GPT, yang memungkinkan pemahaman representasi bahasa secara kontekstual dan mendalam tanpa bergantung pada fitur yang dirancang secara manual [46].

Dalam konteks penelitian ini, NLP dimanfaatkan untuk dua tujuan utama yang saling terintegrasi. Pertama, deteksi ujaran kebencian pada tingkat kalimat (*sentence-level*) pada teks *code-mixing* Indonesia-Inggris, mencakup identifikasi apakah suatu teks mengandung ujaran kebencian beserta skor kepercayaan (*confidence score*) yang menyertai keputusan tersebut. Kedua, identifikasi *hate span*) pada tingkat token digunakan untuk menentukan *span* mana dari teks yang menjadi penyebab ujaran kebencian [24]. Keluaran sistem tidak hanya berupa label biner per kalimat, tetapi juga *hate span* dalam bentuk token tiap kata.

2.3 Arsitektur Transformer

Arsitektur *Transformer* diperkenalkan oleh Vaswani et al. pada tahun 2017 melalui makalah *"Attention is All You Need"* dan menjadi fondasi dominan dalam pengembangan model NLP modern [47]. Sebelum *Transformer*, arsitektur seperti *Recurrent Neural Networks* (RNN) dan *Long Short-Term Memory* (LSTM) menjadi standar utama untuk pemrosesan teks sekuensial. Namun, keduanya memiliki dua keterbatasan mendasar, yaitu kemampuan terbatas dalam menangkap ketergantungan jarak jauh antar token (*long-range dependencies*) dan tidak

dapat diparalelkan selama pelatihan karena sifat pemrosesannya yang sekuensial [46]. *Transformer* mengatasi kedua masalah tersebut melalui mekanisme *self-attention*, yang memungkinkan setiap token untuk secara langsung mengakses dan mempertimbangkan informasi dari token lain dalam kalimat secara paralel [45]. Selain itu, komponen inti dari *Transformer* adalah mekanisme *Scaled Dot-product Attention*. Diberikan tiga matriks masukan yaitu *query* $\mathbf{Q}$, *key* $\mathbf{K}$, dan *value* $\mathbf{V}$, maka keluaran *attention* dihitung secara matematis sebagai berikut [47].

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.1}$$

Faktor $\sqrt{d_k}$ digunakan untuk menskalakan hasil perkalian *dot* agar distribusi gradien tetap stabil selama pelatihan, di mana d_k adalah dimensi vektor *key*. Nilai *softmax* yang dihasilkan merepresentasikan bobot relevansi antar token, yang kemudian digunakan untuk menghasilkan representasi berbobot dari matriks $\mathbf{V}$. Selain itu, mekanisme ini dijalankan secara paralel sebanyak h kali dalam *multi-head attention*, sehingga model dapat menangkap berbagai jenis relasi linguistik secara simultan dari sudut pandang yang berbeda-beda, di mana secara matematis dirumuskan sebagai berikut [46, 47].

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)\,\mathbf{W}^O \tag{2.2}$$

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q,\ \mathbf{K}\mathbf{W}_i^K,\ \mathbf{V}\mathbf{W}_i^V) \tag{2.3}$$

Di sini, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$, dan $\mathbf{W}^O$ adalah matriks parameter yang dapat dipelajari. Setiap *head* memproyeksikan *input* ke subruang dimensi yang berbeda sehingga masing-masing dapat berfokus pada pola relasi yang berbeda pula. BLOOM-560M secara khusus menggunakan 16 *attention head* pada setiap lapisannya. Selain itu, setiap lapisan juga dilengkapi dengan jaringan *feed-forward* yang diterapkan secara independen pada setiap posisi token. BLOOM-560M menggunakan fungsi aktivasi GeLU (*Gaussian Error Linear Unit*) pada lapisan ini, yang secara matematis dirumuskan sebagai berikut [47].

$$\text{FFN}(\mathbf{x}) = \text{GeLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\,\mathbf{W}_2 + \mathbf{b}_2 \tag{2.4}$$

Parameter $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, dan $\mathbf{b}_2$ adalah parameter yang dapat dipelajari selama pelatihan. Sebagai pengganti *positional encoding* berbasis sinusoidal, BLOOM-560M mengadopsi ALiBi (*Attention with Linear Biases*) yang menyuntikkan informasi posisi langsung ke skor *attention* berupa bias linier negatif berdasarkan jarak antar token, tanpa menambah parameter baru pada *embedding*. Setiap lapisan juga dilengkapi dengan *pre-layer normalization* dan koneksi residual untuk memastikan stabilitas pelatihan [47].

Berdasarkan cara model dilatih, arsitektur *Transformer* dapat dibedakan menjadi tiga varian utama. Pertama, model *encoder-only* seperti BERT yang dirancang untuk memahami konteks teks secara dua arah dan unggul pada tugas-tugas klasifikasi serta pelabelan. Kedua, model *decoder-only* seperti GPT yang dirancang untuk menghasilkan teks secara autoregresif dari kiri ke kanan dan unggul pada tugas-tugas generatif. Ketiga, model *encoder-decoder* seperti T5 yang menggabungkan kemampuan pemahaman dan generasi dalam satu arsitektur dan umumnya digunakan untuk tugas-tugas transformasi teks seperti penerjemahan dan peringkasan [46, 47]. Perbedaan mendasar antara ketiga varian ini memiliki implikasi langsung terhadap jenis tugas NLP yang dapat ditangani masing-masing, termasuk dalam konteks deteksi ujaran kebencian dan identifikasi *hate span*.

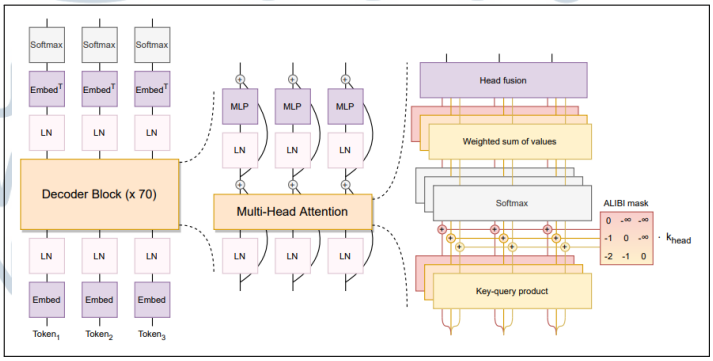
Paradigma *pre-training* dan *fine-tuning* menjadi kunci keberhasilan model berbasis *Transformer*. Pada fase *pre-training*, model dilatih pada korpus teks berskala besar secara tidak terawasi untuk mempelajari representasi bahasa yang kaya dan umum. Selanjutnya, pada fase *fine-tuning*, model diadaptasi untuk tugas spesifik menggunakan dataset berlabel yang lebih kecil [45, 46]. Paradigma ini memungkinkan satu model yang sama untuk digunakan pada berbagai tugas NLP yang berbeda tanpa harus dilatih dari awal, sehingga menghemat sumber daya komputasi secara signifikan sekaligus menghasilkan performa yang lebih baik dibandingkan pendekatan pelatihan dari nol [47].

Dalam konteks penelitian ini, BLOOM-560M dipilih karena mekanisme *self-attention*-nya mampu menangkap ketergantungan jarak jauh antar token, yang krusial untuk mengenali makna ofensif yang bergantung pada konteks kalimat secara keseluruhan, serta Winata et al. menunjukkan bahwa arsitektur *Transformer* multibahasa secara konsisten lebih efektif dibandingkan model monolingual dalam menangani teks *code-mixing*, menjadikan BLOOM-560M yang dilatih pada 46 bahasa alami sebagai fondasi yang lebih tepat dibandingkan model seperti IndoBERT untuk teks *code-mixing* Indonesia-Inggris [9].

2.4 Model BLOOM (BigScience Large Open-science Open-access Multilingual Language Model)

BLOOM merupakan model bahasa multibahasa berbasis *Transformer* yang dikembangkan melalui kolaborasi internasional BigScience dengan melibatkan ratusan peneliti dari berbagai institusi. Model ini dirancang sebagai solusi *open-access* untuk pemrosesan bahasa alami dengan kemampuan memahami dan menghasilkan teks dalam 46 bahasa alami serta 13 bahasa pemrograman. Dengan total 176 miliar parameter dan arsitektur *decoder-only Transformer* yang terdiri dari 70 lapisan, BLOOM merupakan salah satu model bahasa berukuran besar yang tersedia secara terbuka untuk komunitas riset. Berbeda dengan model komersial yang bersifat *proprietary*, BLOOM dikembangkan dengan prinsip keterbukaan sehingga dapat dimanfaatkan secara luas untuk penelitian akademis [27].

Arsitektur BLOOM menggunakan mekanisme *multi-head self-attention* dan *feed-forward neural networks* dengan inovasi ALiBi (*Attention with Linear Biases*) yang meningkatkan kemampuan model dalam menangani sekuens panjang. Arsitektur tersebut dapat divisualisasikan pada Gambar 2.1. Model ini dilatih menggunakan korpus ROOTS (*Responsible Open-science Open-collaboration Text Sources*) yang terdiri dari 1,61 TB data teks multibahasa yang dikurasi dari berbagai sumber seperti buku, artikel ilmiah, dan halaman web. Proses tokenisasi dilakukan dengan *byte-level Byte Pair Encoding* (BPE) dengan *vocabulary* sebesar 250.680 token, yang memungkinkan model menangani kata-kata langka melalui dekomposisi *sub-unit*. Mekanisme *causal language modeling* secara autoregressif memungkinkan model BLOOM menghasilkan teks yang koheren dan sesuai konteks untuk berbagai tugas pemrosesan bahasa alami, termasuk terjemahan mesin, peringkasan teks, *question-answering*, dan generasi kode [27].



Gambar 2.1. Arsitektur model BLOOM

Sumber: [27]

Keunggulan utama model BLOOM, dengan varian 560 juta parameter (BLOOM-560M) untuk penelitian ini terletak pada kemampuannya menangkap nuansa linguistik lintas bahasa secara alami, tanpa memerlukan pemisahan bahasa secara eksplisit. Hal ini sangat relevan mengingat teks *code-mixing* Indonesia-Inggris yang bersifat informal tidak mengikuti batas bahasa yang tegas, sehingga model yang hanya memahami satu bahasa akan mengalami penurunan performa yang signifikan. Selain itu, Wicaksana et al. secara langsung membuktikan bahwa BLOOM menghasilkan akurasi dan *F1-score* yang lebih tinggi pada dataset *code-mixing* Indonesia-Inggris dengan data sintetis [17]. Lebih jauh, *tokenizer* BPE yang dimiliki BLOOM memungkinkan penanganan variasi ortografis informal secara alami melalui dekomposisi *subword*, sehingga tidak diperlukan normalisasi teks yang dapat menghilangkan sinyal linguistik penting pada data ujaran kebencian informal [27, 48]. Karakteristik-karakteristik ini menjadikan BLOOM-560M sebagai model yang paling selaras dengan kebutuhan penelitian ini, yaitu dataset informal, teks *code-mixing* Indonesia-Inggris, dan target deteksi yang mencakup tingkat kalimat dan tingkat token.

2.5 Multi-Task Learning

Multi-Task Learning (MTL) merupakan paradigma pembelajaran mesin di mana sebuah model dilatih secara simultan untuk menyelesaikan dua atau lebih tugas yang berkaitan, dengan berbagi representasi yang dipelajari bersama (*shared representation*) di antara tugas-tugas tersebut [29, 49]. Berbeda dengan pendekatan *single-task* yang mengoptimalkan satu objektif secara terisolasi, MTL memanfaatkan sinyal supervisi dari beberapa tugas sekaligus sebagai mekanisme regularisasi implisit yang mencegah *overfitting* dan mendorong generalisasi yang lebih baik pada data tak terlihat, karena model dipaksa mempelajari representasi yang berguna bagi lebih dari satu tujuan [29]. Efektivitas MTL pada model berbasis *Transformer* telah dibuktikan secara empiris dalam konteks NLP dan *hate speech* di berbagai literatur [50–52].

Secara arsitektural, MTL dapat diimplementasikan melalui dua skema utama [49]. Skema *hard parameter sharing* berbagi seluruh lapisan tersembunyi (*backbone*) di antara semua tugas, sementara setiap tugas memiliki lapisan keluaran (*task-specific head*) yang independen. Skema ini terbukti secara empiris mengurangi risiko *overfitting* secara proporsional dengan jumlah tugas tambahan [29]. Sebaliknya, skema *soft parameter sharing* memberikan setiap tugas model

tersendiri namun meregularisasi parameter antar-model agar tetap berdekatan satu sama lain [49]. Dalam penelitian ini, skema *hard parameter sharing* diterapkan dengan BLOOM-560M sebagai *backbone* bersama yang dioptimalkan untuk dua tugas paralel, yaitu *classification head* untuk deteksi ujaran kebencian di tingkat kalimat dan *token head* untuk identifikasi *hate span* di tingkat token.

2.6 Identifikasi Hate Span dan Pelabelan BIO

Deteksi ujaran kebencian pada tingkat kalimat hanya menghasilkan label biner yang menyatakan suatu teks bersifat ofensif atau tidak. Meskipun berguna, pendekatan ini tidak memberikan informasi mengenai bagian mana dari teks yang menjadi sumber ujaran kebencian, sehingga kemampuan *explainability* sistem menjadi terbatas. Keterbatasan ini mendorong berkembangnya pendekatan identifikasi *hate span* pada tingkat token [24]. Dengan pendekatan ini, sistem tidak hanya menjawab pertanyaan *"apakah teks ini berbahaya?"* tetapi juga *"kata mana yang membuatnya berbahaya?"*, sehingga hasil keluaran sistem dapat diinterpretasikan dan diverifikasi secara lebih konkret.

Identifikasi *hate span* telah diformalkan dalam kompetisi akademis internasional seperti SemEval-2021 Task 5 yang mempopulerkan evaluasi berbasis *token-level F1* dan *strict span matching* sebagai standar pengukuran yang kini diadopsi luas [24]. Subtask HASOC pada FIRE 2023 memperluas cakupan ini dan membuktikan bahwa pendekatan berbasis *Transformer* memberikan hasil yang konsisten lebih baik dibandingkan pendekatan berbasis aturan statis [26]. Integrasi ekstraksi *hate span* dengan klasifikasi ujaran kebencian dalam satu sistem juga terbukti menghasilkan performa yang lebih akurat dan lebih interpretatif dibandingkan sistem klasifikasi tingkat kalimat saja sehingga mendukung desain sistem terpadu pada penelitian ini [25]. Salah satu tantangan utama dalam identifikasi *hate span* adalah anotasi manual pada tingkat token yang jauh lebih mahal dan memakan waktu dibandingkan anotasi tingkat kalimat terutama pada dataset berskala puluhan ribu sampel [53, 54]. Oleh karena itu, penelitian ini mengadopsi pendekatan *weak supervision* berupa TF-IDF *Differential Lexicon* yaitu leksikon ofensif yang dibangun sepenuhnya dari data penelitian sendiri berdasarkan selisih skor TF-IDF antara *corpus hate* dan *corpus non-hate* untuk menghasilkan label BIO secara otomatis tanpa anotasi manual.

Di samping itu, identifikasi *hate span* dimodelkan sebagai tugas pelabelan sekuensial menggunakan skema BIO (*Beginning-Inside-Outside*), yang merupakan

standar yang mapan dalam bidang *Named Entity Recognition* (NER) dan telah terbukti efektif untuk berbagai tugas ekstraksi informasi pada teks [30, 52, 55]. Dalam skema BIO yang diadaptasi untuk identifikasi *hate span*, setiap token dalam kalimat diberi satu dari tiga label, yaitu `B-TOX` (*Beginning of Hate Span*) menandai awal token ujaran kebencian, `I-TOX` (*Inside Hate Span*) menandai *hate-bearing token* lanjutan dalam *span* ujaran kebencian yang sama, dan `O` (*Outside*) menandai token yang tidak termasuk dalam *hate span* manapun. Skema ini memungkinkan representasi *hate span* yang terdiri dari beberapa token berurutan maupun beberapa *hate span* yang terpisah dalam satu kalimat secara tepat dan tidak ambigu.

Skema BIO diformalkan mengikuti kerangka umum pelabelan sekuensial sebagai pemetaan fungsional dari barisan token ke barisan label, di mana diberikan barisan token hasil tokenisasi $x = (x_1, x_2, \ldots, x_n)$, didefinisikan himpunan label $\mathscr{L} = \{\text{O}, \text{B-TOX}, \text{I-TOX}\}$, sehingga sebuah fungsi pelabelan $g : x \mapsto y$ memetakan setiap token x_i ke sebuah label $y_i \in \mathscr{L}$, dan menghasilkan barisan label $y = (y_1, \ldots, y_n)$ dengan $y_i \in \mathscr{L}$ untuk setiap $i \in \{1, \ldots, n\}$ [56]. Tidak seluruh barisan $y \in \mathscr{L}^n$ merupakan barisan yang valid secara struktural. Validitas BIO didefinisikan melalui sebuah relasi transisi yang diperbolehkan dalam barisan y, yaitu $\forall i \in \{2, \ldots, n\} : y_i = \text{I-TOX} \implies y_{i-1} \in \{\text{B-TOX}, \text{I-TOX}\}$. Dengan kata lain, transisi $(\text{O}, \text{I-TOX})$ tidak diperbolehkan, karena token `I-TOX` yang tidak didahului `B-TOX` atau `I-TOX` tidak memiliki awal *span* yang jelas [52, 57].

Tingkat klasifikasi token berbasis lapisan *linear-softmax* semata mengestimasi $P(y_i \mid x)$ secara independen untuk setiap posisi i, sehingga keputusan $\hat{y}_i = \arg\max_{l \in \mathscr{L}} P(y_i = l \mid x)$ tidak mempertimbangkan $\hat{y}_{i-1}$. Lapisan CRF mengatasi hal ini dengan menilai skor gabungan atas seluruh barisan label sekaligus, yang secara matematis diformulasikan sebagai berikut [52, 57].

$$\text{score}(x, y) = \sum_{i=1}^{n} e_i(y_i) + \sum_{i=2}^{n} T(y_{i-1}, y_i) \tag{2.5}$$

$$y^* = \arg\max_{y \in \mathscr{L}^n} \text{score}(x, y) \tag{2.6}$$

Parameter $e_i(y_i)$ adalah skor emisi dari *encoder* BLOOM-560M dan $T(y_{i-1}, y_i)$ adalah skor transisi berpasangan pada matriks transisi $\mathbf{T} \in \mathbb{R}^{|\mathscr{L}| \times |\mathscr{L}|}$ yang dioptimalkan bersama seluruh parameter model. Dikarenakan transisi yang melanggar seperti `O`$\rightarrow$`I-TOX` sangat jarang muncul pada label pelatihan, skor T

untuk transisi tersebut secara empiris menjadi sangat rendah. Pada tahap inferensi, algoritma Viterbi *decoding* mencari probabilitas tertinggi pada Rumus 2.6. Proses ini dilakukan melalui pemrograman dinamis atas seluruh kemungkinan barisan, bukan per-token secara independen, sehingga barisan y^* yang dihasilkan secara empiris cenderung konsisten memenuhi relasi validitas BIO tanpa memerlukan *post-processing* tambahan [52, 57].

2.7 TF-IDF Differential Lexicon

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan ukuran statistik yang mapan dalam bidang *information retrieval* dan pemrosesan bahasa alami untuk mengevaluasi seberapa informatif dan diskriminatif suatu *term* terhadap suatu dokumen dalam konteks sebuah koleksi. Berbeda dari penghitungan frekuensi sederhana, TF-IDF menggabungkan dua properti yang saling melengkapi secara simultan, yaitu seberapa sering suatu *term* muncul dalam dokumen tertentu (*term frequency*), dan seberapa jarang *term* tersebut muncul di seluruh koleksi (*inverse document frequency*). Mekanisme ini secara efektif menekan bobot istilah generik yang hampir selalu muncul di semua dokumen, sambil mengangkat bobot istilah yang khas dan informatif terhadap kelas dokumen tertentu [32, 58]. Komponen *Term Frequency* (TF) mengukur proporsi kemunculan suatu *term* t terhadap seluruh *term* dalam dokumen d (Rumus 2.7). Selain itu, komponen *Inverse Document Frequency* (IDF) mengukur kekhasan suatu *term* t di seluruh koleksi dokumen d (Rumus 2.8) [32, 43, 59].

$$\mathrm{TF}(t,d) = \begin{cases} 1 + \ln\big(\mathrm{tf}(t,d)\big) & \text{jika } \mathrm{tf}(t,d) > 0 \\ 0 & \text{jika } \mathrm{tf}(t,d) = 0 \end{cases} \tag{2.7}$$

$$\mathrm{IDF}(t) = \ln\left(\frac{N+1}{n_t+1}\right) + 1 \tag{2.8}$$

Parameter $\mathrm{tf}(t,d)$ adalah frekuensi kemunculan *term* t dalam dokumen d. Dalam implementasi penelitian ini, nilai TF selanjutnya ditransformasi secara logaritmik (*sublinear scaling*) untuk meredam dominasi token yang sangat sering muncul secara berulang [32, 43]. *Smoothing* pada IDF di Rumus 2.8 mencegah pembagian dengan nol dan memastikan IDF tidak bernilai negatif apabila sebuah *term* muncul di seluruh korpus [58]. Selain itu, parameter N adalah jumlah total

dokumen dan n_t adalah jumlah dokumen yang mengandung *term* t. Skor akhir TF-IDF merupakan perkalian kedua komponen tersebut, sebagaimana dinyatakan sebagai berikut [32, 59].

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \tag{2.9}$$

Term dengan skor TF-IDF tinggi bersifat diskriminatif, yaitu sering muncul dalam dokumen tersebut namun jarang dalam koleksi secara keseluruhan, menjadikannya representasi linguistik yang khas dari kelas dokumen tertentu [33, 58]. TF-IDF telah terbukti menjadi salah satu metode representasi fitur yang paling andal dan konsisten untuk tugas deteksi ujaran kebencian. Studi literatur yang memakai TF-IDF juga menemukan bahwa TF-IDF berbasis *character-level n-gram* secara konsisten memperkuat *classifier* dalam menangani ejaan tidak baku dan kata *slang*, yang relevan dengan karakteristik dataset penelitian ini [60].

Dalam penelitian ini, TF-IDF diterapkan dalam bentuk varian diferensial yang secara spesifik dirancang untuk mengidentifikasi *term* yang bersifat eksklusif terhadap ujaran kebencian. Pendekatan TF-IDF *Differential Lexicon* bekerja dengan menghitung skor TF-IDF setiap *term* secara independen pada dua *corpus* yang saling berlawanan, yaitu *hate corpus* dan *non-hate corpus*. Selisih rata-rata kedua skor tersebut kemudian dihitung sebagai *differential score* (Rumus 2.10) [61, 62].

$$\Delta(t) = \overline{TF\text{-}IDF}_{\text{hate}}(t) \; - \; \overline{TF\text{-}IDF}_{\text{non-hate}}(t) \tag{2.10}$$

Parameter $\overline{TF\text{-}IDF}_{\text{hate}}(t)$ adalah skor TF-IDF rata-rata *term* t pada *hate corpus* dan $\overline{TF\text{-}IDF}_{\text{non-hate}}(t)$ adalah skor TF-IDF rata-rata *term* t pada *non-hate corpus*. Semakin tinggi nilai $\Delta(t)$, semakin eksklusif suatu *term* terhadap teks ujaran kebencian. Hanya *term* dengan $\Delta(t)$ positif dan melampaui ambang persentil adaptif yang masuk ke leksikon akhir [61–63]. Seleksi *term* ke dalam leksikon dilakukan berdasarkan ambang batas persentil yang ditetapkan per subset bahasa, sehingga jumlah entri leksikon dapat menyesuaikan diri secara proporsional dengan ukuran dan karakteristik masing-masing subset [63, 64]. Leksikon dalam penelitian ini juga dibangun dengan mempertimbangkan *unigram* maupun *bigram* untuk menangkap ekspresi ofensif yang bersifat frasa dan tidak dapat diidentifikasi dari kata tunggal saja, dengan ambang batas persentil untuk *bigram* ditetapkan lebih ketat daripada *unigram* guna mengontrol kebisingan yang lebih tinggi pada pola frasa [65–67].

Setelah *offensive lexicon* terbentuk, setiap token dalam dataset yang termasuk dalam leksikon diberi pelabelan BIO secara programatik mengikuti aturan deterministik NER [30, 55, 68]. Secara epistemologis, mekanisme di atas tergolong dalam paradigma *weak supervision*, yakni pendekatan di mana label pelatihan tidak dihasilkan melalui anotasi manual, melainkan melalui fungsi heuristik programatik yang diterapkan secara otomatis. Beberapa literatur menunjukkan bahwa *weak supervision* yang memanfaatkan fungsi heuristik berbasis statistik terbukti efektif dalam menghasilkan label pelatihan yang berkualitas memadai untuk tugas klasifikasi teks, bahkan pada skenario tanpa data berlabel sama sekali dibandingkan menggunakan LLM yang hasilnya tidak konsisten [66, 69, 70].

Aspek implementasi yang menjadi kunci keberhasilan TF-IDF *Differential Lexicon* dalam penelitian ini adalah pembangunan leksikon secara terpisah untuk tiga subset bahasa, yaitu bahasa Indonesia, bahasa Inggris, dan *code-mixing* Indonesia-Inggris. Keputusan ini didasarkan pada pertimbangan bahwa distribusi statistik *term* yang bersifat ofensif berbeda secara signifikan antara ketiga subset tersebut. Kata-kata ujaran kebencian yang khas dalam bahasa Indonesia tidak memiliki relevansi statistik dalam *corpus* bahasa Inggris sehingga skor diferensialnya akan mendekati nol jika dihitung dalam leksikon gabungan. Lebih jauh, pada teks *code-mixing*, variasi ortografis informal seperti *"stupid bgt"* hanya akan muncul secara statistik bermakna di *corpus code-mixing*. Pembangunan leksikon per-subset dengan demikian memastikan bahwa *offensive lexicon* untuk setiap ragam bahasa benar-benar mencerminkan pola leksikal ofensif yang spesifik terhadap konteks linguistik masing-masing, termasuk variasi ortografis tidak baku yang menjadi ciri khas bahasa informal media sosial [5, 60]. Keterbatasan inheren dari *weak supervision*, yaitu bahwa label yang dihasilkan bersifat *noisy* dan tidak seakurat anotasi manual, diakui secara eksplisit sebagai salah satu limitasi penelitian yang menjadi arah penelitian lanjutan [66].

2.8 Hyperparameter Tuning menggunakan Optuna

Performa model BLOOM-560M dalam proses *fine-tuning* sangat dipengaruhi oleh konfigurasi *hyperparameter* yang digunakan selama pelatihan. *Hyperparameter* merupakan parameter eksternal yang nilainya tidak dipelajari secara langsung oleh model, melainkan ditentukan sebelum proses pelatihan dimulai. Parameter seperti *learning rate*, *dropout*, dan bobot komponen *loss* gabungan pada arsitektur *multi-task learning* memiliki pengaruh signifikan

terhadap konvergensi model dan kualitas generalisasi pada seluruh tugas yang dioptimasi secara bersamaan. Penentuan nilai *hyperparameter* secara manual, melalui *Grid Search*, ataupun *Random Search* bersifat tidak efisien karena ruang pencarian yang besar serta adanya interaksi antar-parameter yang kompleks, sehingga pendekatan *automated hyperparameter optimization* (HPO) diperlukan untuk menemukan konfigurasi optimal secara sistematis dan efisien [71, 72].

Optuna merupakan *framework* HPO otomatis berbasis Python yang dirancang dengan prinsip *define-by-run*, yaitu memungkinkan ruang pencarian *hyperparameter* didefinisikan secara dinamis pada setiap *trial* sehingga konstruksi ruang pencarian bersifat fleksibel dan kondisional [71,73]. Tidak seperti pendekatan *define-and-run* yang memerlukan spesifikasi ruang pencarian secara penuh di awal, Optuna mengelola proses HPO melalui konsep *study*, yaitu sesi optimasi yang terdiri dari sejumlah *trial*, di mana setiap *trial* merepresentasikan satu konfigurasi *hyperparameter* yang dievaluasi terhadap fungsi objektif yang telah ditetapkan. Selain itu, Optuna mendukung mekanisme *pruning* untuk menghentikan *trial* yang tidak menjanjikan secara dini berdasarkan hasil evaluasi parsial, sehingga secara keseluruhan mengurangi biaya komputasi yang diperlukan [71].

Inti dari Optuna adalah algoritma *Tree-structured Parzen Estimator* (TPE), yaitu sebuah metode *Bayesian optimization* non-parametrik yang bekerja secara adaptif berdasarkan hasil evaluasi *trial-trial* sebelumnya [71, 74]. Berbeda dengan *Bayesian optimization* berbasis *Gaussian Process* yang memodelkan fungsi objektif secara langsung, TPE membalik pendekatan ini dengan memodelkan distribusi konfigurasi *hyperparameter* yang dikondisikan pada nilai fungsi objektif yang telah diamati, lalu memprioritaskan kandidat yang lebih mungkin berada di kelompok konfigurasi berkinerja tinggi dibandingkan berkinerja rendah. Dengan demikian, TPE secara inheren menyeimbangkan eksplorasi (*exploration*) terhadap area yang belum dijelajahi dan eksploitasi (*exploitation*) terhadap area yang diketahui menjanjikan dalam ruang pencarian [73–75].