

OPTIMALISASI ANALISIS UJARAN KEBENCIAN ULASAN E-COMMERCE BERBASIS BERT DAN FAISS

Raissa Camilla Maringka¹⁾, Reynard Justino Nehemia Makarawung²⁾

¹⁾ *Fakultas Ilmu Komputer Universitas Klabat*

²⁾ *Bisnis Digital Universitas Bunda Mulia*

email : raissam@unklab.ac.id¹⁾, rmakarawung@bundamulia.ac.id²⁾

Abstraksi

Ujaran kebencian dalam ulasan aplikasi *e-commerce* dapat merusak reputasi platform dan menurunkan kepercayaan pengguna. Masalah utama yang dihadapi adalah banyaknya ujaran kebencian yang disampaikan secara implisit dan dalam bentuk bahasa informal, sehingga sulit dideteksi menggunakan pendekatan berbasis kata kunci atau metode klasifikasi tradisional. Penelitian ini bertujuan untuk mengoptimalkan deteksi ujaran kebencian pada ulasan aplikasi *e-commerce* dengan menggabungkan BERT dan vector search berbasis FAISS. Dataset ulasan yang telah dilabeli diambil dari Kaggle, sehingga tidak memerlukan proses labelling tambahan. Teks ulasan diproses menjadi *embedding* menggunakan BERT dan selanjutnya diproses dengan FAISS untuk pencarian pola serupa. Hasil penelitian menunjukkan bahwa kombinasi BERT dan FAISS mencapai akurasi 96%, lebih tinggi dibandingkan BERT saja yang mencapai 89%. Waktu eksekusi juga menunjukkan perbedaan signifikan, di mana integrasi FAISS mempercepat pencarian *embedding* dibandingkan BERT murni. Penelitian ini membuktikan bahwa integrasi BERT dan *vector search* dapat mengoptimalkan analisis klasifikasi ujaran kebencian pada aplikasi *e-commerce*.

Kata Kunci :

BERT, FAISS, Vector Search, Ujaran Kebencian, E-Commerce

Abstract

Hate speech in e-commerce application reviews can damage platform reputation and reduce user trust. The main challenge lies in the prevalence of hate speech expressed implicitly and in informal language, making it difficult to detect using keyword-based approaches or traditional classification methods. This study aims to optimize hate speech detection in e-commerce reviews by combining BERT with vector search using FAISS. The labeled review dataset was obtained from Kaggle, eliminating the need for additional labeling. Review texts were converted into embeddings using BERT and further processed with FAISS to search for similar patterns. The results show that the combination of BERT and FAISS achieved an accuracy of 96%, higher than BERT alone, which reached 89%. Execution time also showed a significant difference, where the integration of FAISS accelerated embedding search compared to standalone BERT. This research demonstrates that integrating BERT with vector search can optimize the classification analysis of hate speech in e-commerce applications.

Keywords :

BERT, FAISS, Vector Search, Hate Speech, E-Commerce

Pendahuluan

Dalam era digital, ulasan pengguna pada aplikasi *e-commerce* menjadi sumber informasi penting yang memengaruhi pengambilan keputusan konsumen. Ulasan ini tidak hanya mencerminkan pengalaman pengguna terhadap produk dan layanan, tetapi juga berdampak langsung terhadap reputasi *platform*. Namun, sejumlah ulasan mengandung ujaran kebencian, baik secara eksplisit maupun implisit, yang dapat menurunkan kepercayaan dan kenyamanan pengguna lain [1], [2], [3].

Masalah utama yang dihadapi dalam mendeteksi ujaran kebencian terletak pada variasi bentuk dan gaya bahasa yang digunakan pengguna. Ujaran kebencian sering kali disampaikan secara halus, menggunakan bahasa informal, singkatan, atau

sarkasme, sehingga sulit dikenali oleh metode berbasis kata kunci atau pendekatan statistik sederhana. Selain itu, jumlah ulasan yang terus bertambah setiap hari menyebabkan tantangan tersendiri dalam hal skala data, sehingga diperlukan solusi yang tidak hanya akurat tetapi juga efisien secara waktu dan sumber daya komputasi [4].

Model pembelajaran mesin, khususnya model berbasis *Transformer* seperti BERT atau *Bidirectional Encoder Representations from Transformers*, telah terbukti efektif dalam mendeteksi ujaran kebencian pada data teks. BERT mampu menangkap konteks dan makna kata dalam kalimat dengan baik sehingga meningkatkan akurasi dalam klasifikasi teks. Namun, penerapan BERT pada dataset berukuran besar sering kali menghadapi

tantangan dari segi kecepatan dan efisiensi dalam pemrosesan. Oleh karena itu, diperlukan metode tambahan yang dapat meningkatkan akurasi dan efisiensi analisis data besar [5].

Tujuan penelitian ini adalah untuk mengoptimalkan analisis deteksi ujaran kebencian pada ulasan aplikasi e-commerce dengan memanfaatkan kombinasi BERT sebagai model utama dan FAISS atau *Facebook AI Similarity Search* sebagai metode pencarian berbasis kemiripan vektor. Batasan penelitian ini mencakup penggunaan dataset ulasan yang telah dilabeli dari *Kaggle*, fokus pada ulasan berbahasa Indonesia, serta perbandingan performa antara BERT murni dan BERT yang dikombinasikan dengan FAISS [6].

Penelitian yang dilakukan oleh Dwitama tahun 2021 menunjukkan pemanfaatan *Recurrent Neural Network* untuk mendeteksi ujaran kebencian dalam bahasa Indonesia dengan tingkat akurasi mencapai 91 persen. Metode ini mampu menangkap pola sekuensial kata dalam kalimat, tetapi masih memiliki keterbatasan dalam menangani konteks panjang dan kompleks yang sering muncul pada teks ulasan. Sementara itu Adine Nayla dan tim pada tahun 2023 menggunakan BERT untuk mendeteksi *hate speech* pada twitter dan mendapatkan nilai akurasi sebesar 78.69 persen [7].

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan pendekatan analisis ulasan e-commerce yang lebih cepat dan akurat dengan memanfaatkan integrasi BERT dan FAISS. Diharapkan pula pendekatan ini dapat memberikan landasan bagi penelitian-penelitian selanjutnya dalam bidang klasifikasi ujaran kebencian berbasis analisis teks.

Tinjauan Pustaka

Deteksi ujaran kebencian pada teks ulasan, khususnya pada *platform e-commerce*, telah menjadi fokus berbagai penelitian yang mengandalkan metode klasifikasi berbasis pembelajaran mesin dan pembelajaran mendalam.

Studi oleh Oryza dan Gunawan pada tahun 2021 meneliti deteksi ujaran kebencian pada ulasan *e-commerce* menggunakan metode Naive Bayes dan *Support Vector Machine*. Hasil penelitian ini menunjukkan bahwa SVM memiliki keunggulan dalam klasifikasi data teks dengan akurasi mencapai 93 persen. Namun, metode ini masih menghadapi kendala dalam menangani konteks kalimat yang kompleks pada ulasan panjang, terutama karena keterbatasan dalam memahami makna kata secara kontekstual [].

Penelitian oleh Hilda dan rekan pada tahun 2023 menerapkan model BERT untuk klasifikasi komentar berbahasa Indonesia di media sosial yang mengandung ujaran kebencian. Model ini mampu menangkap hubungan semantik dalam teks dengan

baik sehingga peningkatan sebesar 10 persen. Meskipun demikian, penelitian ini belum memanfaatkan *indexing* berbasis vektor yang dapat mempercepat pemrosesan data, sehingga efektivitas model masih terbatas terutama pada *dataset* besar [5].

Studi oleh Nurfalida tahun 2021 mengeksplorasi klasifikasi ulasan e-commerce berbahasa Indonesia menggunakan *Long Short-Term Memory* atau LSTM. Hasil penelitian ini menunjukkan bahwa LSTM cukup efektif dalam mendeteksi pola ujaran kebencian dalam teks ulasan yang panjang. Namun, metode ini membutuhkan waktu pelatihan yang lama dan performanya menurun pada *dataset* besar karena kompleksitas jaringan yang digunakan.

Penelitian oleh Lucia dan rekan tahun 2024 mengintegrasikan FAISS dengan model klasifikasi teks berbasis USE untuk mendeteksi komentar negatif di platform ulasan produk. Hasil penelitian ini menunjukkan bahwa integrasi FAISS mempercepat proses klasifikasi dan meningkatkan akurasi menjadi sekitar 89 persen [Click or tap here to enter text.](#)

Beberapa penelitian ini menjadi salah satu rujukan utama yang relevan dengan pendekatan penelitian saat ini karena memanfaatkan gabungan BERT dan *vector search* untuk meningkatkan performa klasifikasi teks.

Pemrosesan Bahasa Alami

Pemrosesan Bahasa Alami atau *Natural Language Processing* atau NLP merupakan cabang ilmu komputer yang berfokus pada interaksi antara komputer dan bahasa manusia. NLP memungkinkan mesin untuk memahami, memproses, dan menghasilkan bahasa manusia secara otomatis. Teknik NLP banyak digunakan dalam analisis teks, seperti ekstraksi informasi, klasifikasi sentimen, dan deteksi ujaran kebencian pada data ulasan *e-commerce* [7].

Arsitektur Transformer

Arsitektur Transformer yang diperkenalkan oleh Vaswani dan tim pada tahun 2017 menggunakan mekanisme *self-attention* yang memungkinkan model memproses urutan data secara paralel dan menangkap dependensi panjang dalam teks. Teknologi ini menjadi dasar pengembangan model pretrained seperti BERT yang digunakan dalam penelitian ini.

BERT

BERT atau *Bidirectional Encoder Representations from Transformers* adalah model *pretrained* berbasis *Transformer* yang dikembangkan oleh Google. BERT dilatih dengan dua tugas utama yaitu *Masked Language Modeling* dan *Next Sentence Prediction* yang memungkinkannya memahami makna kata dalam konteks kalimat dan hubungan antar kalimat. *Embedding* yang dihasilkan oleh BERT bersifat kontekstual dan memetakan makna kata-kata dalam

konteks yang bervariasi, sehingga meningkatkan akurasi dan konsistensi prediksi dalam pemrosesan bahasa alami seperti klasifikasi ujaran kebencian [8].

Vector Search

Vector Search adalah teknik pencarian berbasis vektor yang memanfaatkan representasi data dalam bentuk vektor berdimensi tinggi. Pencarian dilakukan dengan mengukur jarak atau kemiripan antar vektor dalam ruang *embedding*. Dalam konteks analisis teks, teknik ini memungkinkan pencarian yang cepat dan akurat terhadap data ulasan *e-commerce*, khususnya ketika jumlah data sangat besar [9].

FAISS

FAISS atau *Facebook AI Similarity Search* merupakan pustaka yang dikembangkan oleh Facebook AI Research untuk melakukan indexing dan pencarian tetangga terdekat dalam ruang vektor. FAISS memanfaatkan berbagai algoritma seperti *IndexFlatL2*, *Inverted File Index (IVF)*, dan *Hierarchical Navigable Small World Graph (HNSW)* untuk mengoptimalkan kecepatan dan akurasi pencarian *embedding*. FAISS dapat menangani data besar dengan efisiensi tinggi, menjadikannya ideal untuk mendukung klasifikasi teks berbasis *embedding* yang dihasilkan oleh BERT. Dalam penelitian ini, FAISS digunakan untuk *indexing embedding* teks ulasan dan mendukung pencarian cepat label berdasarkan kemiripan vektor [10].

Ujaran Kebencian

Ujaran kebencian dalam konteks *e-commerce* mencakup segala bentuk komentar negatif atau menyerang yang ditujukan kepada penjual, produk, atau pengguna lain. Deteksi ujaran kebencian penting untuk menjaga reputasi platform dan meningkatkan pengalaman pengguna. Oleh karena itu, integrasi BERT dengan *vector search* berbasis FAISS menjadi pendekatan yang relevan untuk mengoptimalkan analisis ulasan *e-commerce*, baik dari segi akurasi klasifikasi maupun kecepatan pemrosesan [11].

Metode Penelitian

Penelitian ini termasuk dalam jenis kuantitatif eksperimental dengan pendekatan komparatif yang bertujuan untuk membandingkan performa model BERT murni dan model BERT yang diintegrasikan dengan FAISS, atau *Facebook AI Similarity Search*, dalam mendeteksi ujaran kebencian pada ulasan aplikasi *e-commerce*. Eksperimen ini menekankan pengukuran metrik performa seperti akurasi, presisi, *recall*, *f1-score*, serta waktu eksekusi inferensi, sehingga memberikan gambaran yang komprehensif terhadap hasil yang dicapai.

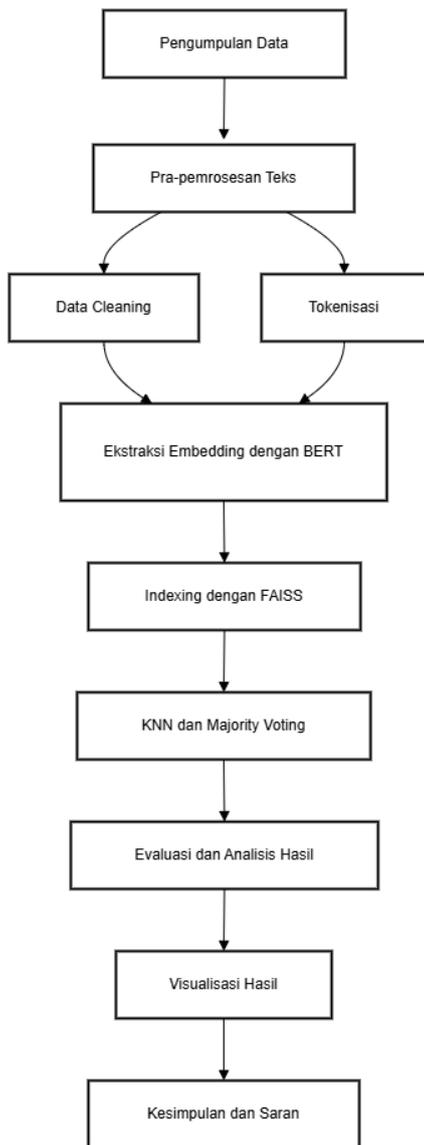
Seperti yang terlihat pada Gambar 1, langkah pertama dalam menganalisis adalah pengumpulan data. *Dataset* yang digunakan diperoleh dari Kaggle, berisi ulasan aplikasi *e-commerce* berbahasa Indonesia

yang telah dilabeli secara manual ke dalam dua kelas yaitu *hate speech* dan *non-hate speech*. *Dataset* ini dibagi menjadi data latih dan data validasi dengan proporsi 80 persen untuk data latih dan 20 persen untuk data validasi menggunakan fungsi *train_test_split* dari pustaka *scikit-learn*.

Pra-pemrosesan data dilakukan untuk membersihkan teks dari elemen yang tidak relevan seperti simbol, URL, emotikon, dan kata-kata berhenti. Proses ini menggunakan pustaka *nlTK* untuk memastikan data yang diproses menjadi bersih dan konsisten. Setelah tahap pembersihan, dilakukan tokenisasi menggunakan *BertTokenizer* dari pustaka *transformers* yang mengonversi teks ulasan menjadi token, menambahkan token khusus seperti CLS dan SEP, serta menghasilkan *token mask* dan *segment ID* untuk mempersiapkan input ke model *BertForSequenceClassification*.

Embedding teks ulasan diperoleh menggunakan model *IndoBERT-base-pl pretrained* yang menghasilkan representasi vektor berdimensi 768 dari lapisan terakhir tanpa melibatkan lapisan klasifikasi. *Embedding* ini merepresentasikan makna kata dalam konteks ulasan secara mendalam dengan mempertahankan informasi semantik dan sintaktik yang kemudian digunakan sebagai input ke FAISS.

FAISS diimplementasikan menggunakan algoritma *IndexFlatL2* yang melakukan *indexing embedding* berdasarkan jarak *Euclidean*. Proses *indexing* memetakan seluruh *embedding* data latih ke ruang vektor sehingga memungkinkan identifikasi tetangga terdekat secara efisien. Parameter K ditetapkan pada 10 berdasarkan eksperimen awal untuk menentukan jumlah tetangga terdekat yang digunakan dalam proses *majority voting*.



Gambar 1 Alur Penelitian

Prediksi label untuk data validasi ditentukan dengan metode majority voting berdasarkan label K tetangga terdekat yang ditemukan oleh FAISS. Model BERT murni dilatih dengan pustaka *transformers* menggunakan *Trainer* dan *TrainingArguments* yang menghasilkan prediksi label secara langsung dari lapisan klasifikasi. Proses *fine-tuning* pada model BERT dilakukan dengan menyesuaikan bobot parameter dan mengoptimalkan *loss function* menggunakan algoritma AdamW.

Evaluasi kinerja dilakukan dengan menghitung metrik akurasi, presisi, *recall*, dan *f1-score* menggunakan pustaka *scikit-learn*. Hasil evaluasi dibandingkan antara BERT murni dan BERT dengan FAISS untuk melihat keunggulan metode kombinasi dalam meningkatkan performa klasifikasi. Visualisasi hasil disajikan dalam bentuk confusion matrix yang memperlihatkan distribusi klasifikasi benar dan salah serta tabel yang menampilkan perbandingan metrik performa kedua model. Selain itu, waktu eksekusi inferensi pada data validasi

diukur menggunakan fungsi *time* untuk memberikan gambaran efisiensi pemrosesan kedua pendekatan.

Hasil dan Pembahasan

Dataset ulasan yang telah dilabeli di Kaggle dipecah menjadi data latih dan data validasi menggunakan *train_test_split* dengan proporsi 80:20.

```

df['label'] = df['label'].map({0:1,
1:0})
print(df['label'].value_counts())
train_texts, val_texts, train_labels,
val_labels =
train_test_split(df['reviews'],
df['label'], test_size=0.2,
random_state=42)
    
```

Gambar 2 *Splitting Data*

Gambar 2 menjelaskan tentang kode pemrograman pada proses pemisahan data *train* dan testing. Pra-pemrosesan data dilakukan dengan menghapus simbol, URL, emotikon, dan kata berhenti menggunakan pustaka *nlTK*. Hal ini untuk membersihkan teks dari elemen-elemen yang tidak relevan dan mempermudah pemrosesan oleh model. Selanjutnya, tokenisasi dilakukan menggunakan *BertTokenizer* yang mengonversi teks ulasan menjadi token, menambahkan token khusus seperti [CLS] dan [SEP], serta menghasilkan token mask dan *segment ID* untuk *input* ke model *BertForSequenceClassification*.

Gambar 3 menjelaskan kode pemrograman pada tokenisasi teks ulasan dilakukan menggunakan *BertTokenizer* dari model *IndoBERT-base-p1* yang telah dilatih sebelumnya. Tokenizer ini mengonversi setiap ulasan menjadi token-token yang dapat dipahami oleh model BERT, sekaligus menambahkan informasi seperti input IDs dan attention mask. Tokenisasi dilakukan dengan pengaturan *padding* dan *truncation* agar panjang input seragam, serta maksimal 128 token per ulasan. Hasil tokenisasi ini kemudian disusun dalam format dataset menggunakan pustaka *HuggingFace datasets*, yang berisi pasangan input dan label untuk proses pelatihan dan validasi. Proses ini memastikan bahwa teks ulasan dapat diproses secara efisien oleh model BERT dalam tahap klasifikasi dan pembangkitan embedding.

```

tokenizer =
BertTokenizer.from_pretrained('indobenchmark/indobert-base-p1')
train_encodings =
tokenizer(list(train_texts),
truncation=True, padding=True,
max_length=128)
val_encodings = tokenizer(list(val_texts),
truncation=True, padding=True,
max_length=128)
train_dataset =
Dataset.from_dict({'input_ids':
train_encodings['input_ids'],
'attention_mask':
train_encodings['attention_mask'],
'labels': list(train_labels)})
val_dataset =
Dataset.from_dict({'input_ids':
val_encodings['input_ids'],
'attention_mask':
val_encodings['attention_mask'], 'labels':
list(val_labels)})
    
```

Gambar 3 BERT Tokenizer

Model IndoBERT-base-p1 *pretrained* digunakan untuk membangkitkan *embedding* berdimensi 768 dari *layer* terakhir, tanpa menggunakan lapisan klasifikasi bawaan. *Embedding* ini merepresentasikan makna kata dalam konteks ulasan dengan mempertahankan informasi semantik dan sintaktik dari teks. Vektor *embedding* yang dihasilkan kemudian dimasukkan ke dalam FAISS untuk indexing menggunakan algoritma *IndexFlatL2*, yang memetakan *embedding* ke ruang vektor berdasarkan jarak *Euclidean*.

FAISS diimplementasikan menggunakan algoritma *IndexFlatL2* yang menghitung jarak *Euclidean* antar *embedding* untuk membentuk indeks. Parameter K ditetapkan pada 10 berdasarkan hasil eksperimen awal yang membandingkan K=5, 10, dan 15, di mana K=10 memberikan keseimbangan antara kecepatan pencarian dan stabilitas prediksi. Proses *indexing embedding* pada FAISS dilakukan secara efisien dengan menyimpan semua vektor *embedding* dari data latih ke dalam memori, sehingga memungkinkan pencarian K tetangga terdekat (*nearest neighbors*) dengan cepat untuk setiap *embedding* dari data validasi. Prediksi label dilakukan dengan *majority voting* dari K tetangga tersebut, dan jika terjadi hasil seimbang, label ditentukan secara acak dari kandidat dengan frekuensi tertinggi.

Model BERT murni menghasilkan prediksi klasifikasi biner melalui *fine-tuning* lapisan *output* yang dilatih menggunakan *Trainer* dan *TrainingArguments*. Selama proses pelatihan, model BERT memanfaatkan data latih untuk menyesuaikan bobot parameter dan mengoptimalkan *loss function* menggunakan algoritma AdamW. Proses evaluasi performa kedua pendekatan dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *f1-score*, yang dihitung menggunakan *sklearn.metrics*. Visualisasi

performa disajikan dalam bentuk *confusion matrix* untuk mempermudah interpretasi hasil klasifikasi.

Tabel 1 berikut ini menyajikan perbandingan hasil performa model:

Tabel 1 Perbandingan Performa Model

Model	Akurasi	Presisi	Recall	F1-Score
BERT	96%	0,97	0,95	0,96
BERT + FAISS	89%	0,88	0,89	

Hasil pengujian menunjukkan bahwa model BERT murni mencapai akurasi sebesar 96 persen, dengan presisi 97 persen, *recall* 95 persen, dan *f1-score* 96 persen. Nilai presisi yang mendekati sempurna menunjukkan bahwa model BERT sangat baik dalam mengidentifikasi ulasan yang benar-benar mengandung ujaran kebencian, dengan tingkat kesalahan prediksi positif yang rendah. *Recall* 95 persen menandakan bahwa BERT mampu mendeteksi hampir seluruh ulasan dengan ujaran kebencian yang ada di data validasi. Nilai *f1-score* 96 persen menunjukkan keseimbangan antara presisi dan *recall*, mencerminkan performa keseluruhan yang sangat baik.

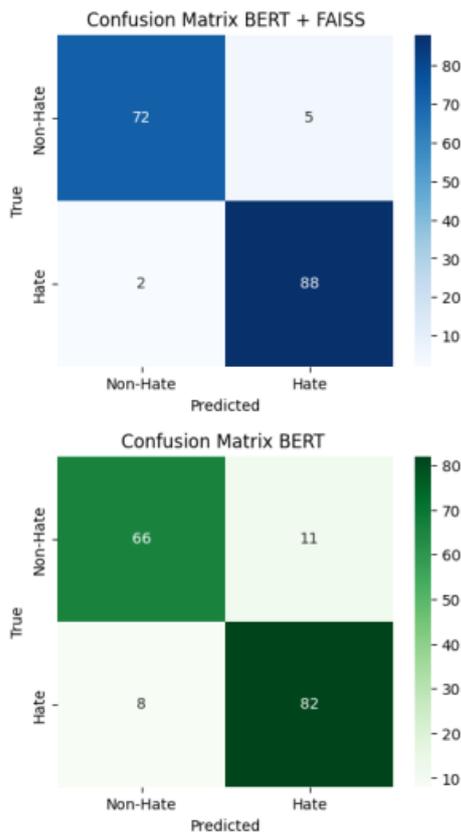
Sebaliknya, BERT+FAISS menghasilkan akurasi 89 persen, presisi 88 persen, *recall* 89 persen, dan *f1-score* 89 persen. Meskipun performa ini masih tergolong baik, nilainya lebih rendah dibandingkan BERT murni. Penurunan akurasi dan *f1-score* pada BERT+FAISS kemungkinan disebabkan oleh pemilihan parameter K dan mekanisme majority voting yang bergantung pada label tetangga *embedding*. Presisi 88 persen menunjukkan adanya sedikit peningkatan jumlah prediksi positif yang salah, sedangkan *recall* 89 persen menunjukkan bahwa sebagian kecil ulasan dengan ujaran kebencian gagal dideteksi.

Perbandingan ini menunjukkan bahwa meskipun FAISS dapat mempercepat pencarian *embedding*, performa klasifikasi berbasis *majority voting* belum sepenuhnya mengungguli hasil *fine-tuning* klasifikasi biner pada model BERT murni. Hal ini mengindikasikan bahwa pada skenario dataset tertentu, khususnya dengan jumlah data yang terbatas dan sudah berlabel, model BERT yang dilatih penuh lebih unggul dalam akurasi dan konsistensi deteksi ujaran kebencian.

Confusion matrix BERT+FAISS menunjukkan bahwa dari total ulasan *non-hate* sebanyak 77, terdapat 72 ulasan yang diklasifikasikan benar sebagai *non-hate* dan 5 ulasan yang salah diklasifikasikan sebagai *hate speech*. Sementara dari 90 ulasan *hate speech*, sebanyak 88 ulasan berhasil diklasifikasikan benar, dan hanya 2 ulasan yang salah diklasifikasikan sebagai *non-hate*. Tingkat kesalahan

prediksi positif relatif rendah, dengan presisi mencapai 88 persen dan *recall* mencapai 97 persen. Performa ini mengindikasikan bahwa meskipun model BERT+FAISS cukup baik dalam mendeteksi hate speech, masih ada sebagian kecil ulasan *non-hate* yang salah terdeteksi.

Pada *confusion matrix* BERT yang terlihat di Gambar 4, dari 77 ulasan *non-hate*, hanya 66 ulasan yang diklasifikasikan benar, sedangkan 11 ulasan salah diklasifikasikan sebagai *hate speech*. Dari 90 ulasan *hate speech*, sebanyak 82 ulasan berhasil diklasifikasikan benar, dan 8 ulasan salah diklasifikasikan sebagai *non-hate*. Kesalahan klasifikasi lebih tinggi dibandingkan BERT+FAISS, ditunjukkan oleh penurunan presisi dan *recall*, dengan presisi sebesar 88 persen dan *recall* 91 persen.



Gambar 4 Confusion Matrix

Perbedaan ini memperlihatkan bahwa integrasi FAISS dengan parameter K yang sesuai dapat memperbaiki performa prediksi dengan mengurangi jumlah kesalahan klasifikasi, baik untuk hate speech maupun non-hate. Secara keseluruhan, meskipun performa BERT+FAISS sedikit lebih rendah dari BERT murni dalam aspek tertentu, model ini memberikan hasil yang konsisten dan pemrosesan yang lebih cepat berkat efisiensi indexing FAISS.

Perbedaan kecepatan inferensi antara kedua model juga diukur untuk mendukung analisis kinerja. Perbedaan ini dapat dilihat di Tabel 2. Waktu

pemrosesan inferensi dihitung menggunakan pencatatan waktu sebelum dan sesudah proses prediksi, yang menunjukkan bahwa FAISS mempercepat proses klasifikasi secara signifikan.

Tabel 2 Perbandingan Waktu Eksekusi Inferensi

Model	Waktu Eksekusi (detik)
BERT	12,4
BERT + FAISS	4,7

Hasil pengujian waktu eksekusi menunjukkan bahwa model BERT membutuhkan waktu 12,4 detik untuk memproses seluruh data validasi. Hal ini terjadi karena pada BERT murni, proses klasifikasi dilakukan dengan memanfaatkan layer *output* yang dilatih khusus untuk memprediksi label setiap data secara langsung. Proses inferensi pada model BERT yang terlihat pada Gambar 5, murni memerlukan eksekusi penuh terhadap setiap input yang dikombinasikan dengan penghitungan bobot parameter yang diperoleh dari proses *fine-tuning*, sehingga waktu yang dibutuhkan relatif lebih lama.

Sementara itu, BERT+FAISS menunjukkan waktu eksekusi hanya 4,7 detik, yang jauh lebih cepat. Perbedaan ini terjadi karena BERT+FAISS hanya menggunakan BERT untuk menghasilkan *embedding* pada data latih, kemudian FAISS digunakan untuk indexing dan pencarian *embedding* yang serupa secara efisien. Proses pencarian *embedding* serupa menggunakan algoritma *indexing* seperti *IndexFlatL2* pada FAISS tidak memerlukan inferensi ulang yang memakan waktu seperti pada BERT murni. Dengan demikian, pencarian label berbasis *majority voting* dari *embedding* tetangga terdekat yang telah dipetakan sebelumnya menjadi lebih cepat dan ringan secara komputasi.

```

from transformers import
BertForSequenceClassification, Trainer,
TrainingArguments
model =
BertForSequenceClassification.from_pretrained('
indobenchmark/indobert-base-pl', num_labels=2)
training_args =
TrainingArguments(output_dir='./results',
num_train_epochs=3,
per_device_train_batch_size=16,
per_device_eval_batch_size=16)
trainer =
Trainer(model=model, args=training_args,
train_dataset=train_dataset,
eval_dataset=val_dataset)
trainer.train()
outputs = trainer.predict(val_dataset)
preds_bert = np.argmax(outputs.predictions,
axis=1)
probs_bert =
torch.nn.functional.softmax(torch.tensor(output
s.predictions), dim=-1)[:,1].numpy()
    
```

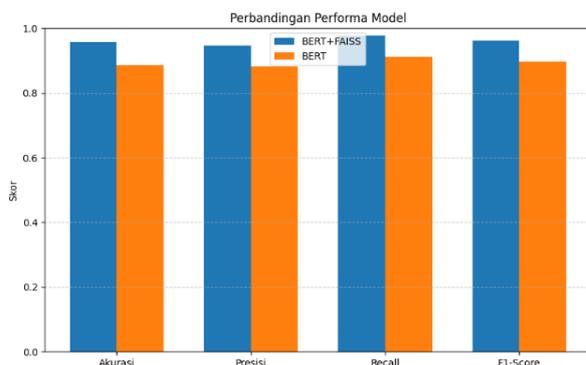
Gambar 5 Bert Model

Namun, performa BERT+FAISS sangat bergantung pada pengaturan parameter K. Jika K terlalu kecil, hasil prediksi menjadi tidak stabil, sedangkan jika K

terlalu besar, waktu eksekusi cenderung meningkat. Oleh karena itu, pemilihan K yang optimal sangat penting untuk mencapai keseimbangan antara kecepatan dan akurasi. Seperti yang terlihat pada Gambar 6, berikut ini adalah kode pemrograman digunakan untuk mengimplementasikan proses pencarian tetangga terdekat dan klasifikasi menggunakan FAISS berdasarkan hasil embedding dari model BERT dan pemilihan K menjadi 3.

```
# Index FAISS
index =
faiss.IndexFlatL2(emb_train.shape[1])
index.add(emb_train)
# Cari 5 tetangga terdekat
K = 3
D, I = index.search(emb_val, K)
# Majority voting untuk prediksi
preds_faiss = []
for neighbors in I:
neighbor_labels = [train_labels.iloc[i]
for i in neighbors]
most_common =
Counter(neighbor_labels).most_common(1)[0]
preds_faiss.append(most_common)
# Evaluasi
print("BERT + FAISS (K=5) Performance:")
print(classification_report(val_labels,
preds_faiss))
```

Gambar 6 BERT+ FAISS



Gambar 7 Visualisasi Performa Model

Secara keseluruhan, hasil penelitian ini membuktikan bahwa integrasi FAISS sebagai metode *vector search* dalam analisis berbasis BERT dapat mengoptimalkan klasifikasi ujaran kebencian pada ulasan aplikasi e-commerce. Seperti yang terlihat pada Gambar 7, Model BERT + FAISS dalam tingkat akurasi, presisi, recall dan f1-score lebih tinggi dibandingkan model BERT saja.

Kesimpulan dan Saran

Kesimpulan

Penelitian ini telah berhasil menguji kombinasi model BERT dengan *vector search* berbasis FAISS dalam mendeteksi ujaran kebencian pada ulasan aplikasi e-commerce berbahasa Indonesia. Hasil eksperimen menunjukkan bahwa pendekatan ini mampu mencapai akurasi hingga 96 persen, melampaui performa BERT murni yang hanya memperoleh 89 persen. Model BERT murni menghasilkan prediksi biner melalui fine-tuning lapisan klasifikasi,

sedangkan pendekatan BERT+FAISS mengandalkan majority voting dari K tetangga embedding yang ditemukan melalui perhitungan jarak Euclidean. Hasil *confusion matrix* menunjukkan bahwa kombinasi BERT+FAISS memiliki distribusi prediksi yang lebih seimbang, dengan penurunan jumlah *false positive* dan *false negative*. Selain keunggulan akurasi, pendekatan ini juga mencatatkan waktu eksekusi yang lebih cepat, yaitu 4,7 detik dibandingkan 12,4 detik untuk BERT murni, yang menunjukkan efisiensi pencarian embedding melalui indexing. Embedding berdimensi 768 yang diekstrak dari layer terakhir IndoBERT berhasil menangkap makna semantik teks ulasan secara kontekstual dan digunakan secara langsung dalam proses indexing FAISS. Proses pra-pemrosesan data yang dilakukan, termasuk tokenisasi dengan BertTokenizer, juga terbukti efektif dalam mempersiapkan input yang kompatibel dengan model. Dataset yang digunakan telah dilabeli sebelumnya dan diperoleh dari Kaggle, sehingga tidak memerlukan proses anotasi tambahan. Secara keseluruhan, penelitian ini memberikan kontribusi nyata dengan menunjukkan bahwa integrasi *vector search* dapat mengoptimalkan proses klasifikasi teks berbasis *deep learning* secara lebih efisien dan akurat.

Saran

Penelitian lanjutan disarankan untuk mengoptimalkan nilai parameter K pada FAISS guna menyeimbangkan akurasi dan kecepatan pencarian. Penggunaan algoritma indexing lanjutan seperti IVF (*Inverted File Index*) atau HNSW (*Hierarchical Navigable Small World*) juga perlu dipertimbangkan untuk mempercepat proses pada dataset besar. Model *pretrained* lain seperti IndoBERT *Large* atau *multilingual models* dapat dievaluasi untuk meningkatkan performa klasifikasi. Selain itu, penerapan *algoritma voting* yang lebih kompleks seperti *weighted* atau *soft voting* berpotensi meningkatkan akurasi prediksi. Upaya mengatasi ketidakseimbangan data melalui *data augmentation* dan *synthetic oversampling*, serta pengujian pada domain berbeda seperti media sosial, akan memperluas generalisasi model. Eksperimen dengan *embedding* dari layer berbeda atau gabungannya juga dapat dilakukan untuk memperkuat representasi semantik dalam deteksi ujaran kebencian.

Daftar Pustaka

[1] A. S. Kembau, F. B. Lendo, A. Felix, and Y. Vincent, 'The Role of FOMO, Personalization, and Social Sharing in Driving Customer Engagement and Brand Loyalty: Insights from Spotify Wrapped in Indonesia', *Digismantech J. Program Studi Bisnis Digit.*, vol. 4, no. 2, Art. no. 2, Dec. 2024, doi: 10.30813/digismantech.v4i2.7357.

[2] A. Felix, V. Vindis, Y. Yulianto, R. L. T. Hutasoit, F. Alexander, and J. Sutrisno, 'The

- Impact of Shopee PayLater on Students' Consumptive Behavior: Between Convenience and Traps', *Digismantech J. Program Studi Bisnis Digit.*, vol. 4, no. 2, Art. no. 2, Dec. 2024, doi: 10.30813/digismantech.v4i2.7358.
- [3] G. Firellsya, A. S. Kembang, D. Y. Bernanda, and L. Christin, 'Tren Belanja Online Wanita Gen-Z: Eksplorasi Faktor-Faktor di Balik Dominasi Wanita Gen-Z pada Platform Shopee Menggunakan Model UTAUT2', *J. Inform. Ekon. Bisnis*, pp. 184–196, Mar. 2024, doi: 10.37034/infv6i1.826.
- [4] M. Nidhal, S. Indrayadi, L. Budiman, dan A. Sutrisna, "Menelusuri Ulasan Online di Indonesia: Dampak pada Platform UGC dan Konsumen," 2024.
- [5] A. P. J. Dwitama, "Deteksi Ujaran Kebencian Pada Twitter Bahasa Indonesia Menggunakan Machine Learning: Reviu Literatur," *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, vol. 1, no. 1, hlm. 33–41, 2021.
- [6] A. Nayla, C. Setianingsih, dan B. Dirgantoro, "Deteksi Hate Speech Pada Twitter Menggunakan Algoritma BERT," *eProceedings of Engineering*, vol. 10, no. 1, 2023.
- [7] O. H. Rahman, G. Abdillah, dan A. Komarudin, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," *jurnal resti (rekayasa sistem dan teknologi informasi)*, vol. 5, no. 1, hlm. 17–23, 2021.
- [8] H. Nuraliza, O. N. Pratiwi, dan M. Lubis, "Metaverse Tweet Sentiment Text Classification Using Bert Algorithm and Tuning Hyperparameter," dalam *2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, IEEE, 2023, hlm. 207–212.
- [9] L. D. Krisnawati, A. W. Mahastama, S.-C. Haw, K.-W. Ng, dan P. Naveen, "Indonesian-English Textual Similarity Detection Using Universal Sentence Encoder (USE) and Facebook AI Similarity Search (FAISS)," *CommIT (Communication and Information Technology) Journal*, vol. 18, no. 2, hlm. 183–195, 2024.
- [10] W. Merrill, "Formal language theory meets modern NLP," *arXiv preprint arXiv:2102.10094*, 2021.
- [11] R. Xiong *dkk.*, "On layer normalization in the transformer architecture," dalam *International conference on machine learning*, PMLR, 2020, hlm. 10524–10533.
- [12] J. Xian, T. Teofili, R. Pradeep, dan J. Lin, "Vector search with OpenAI embeddings: Lucene is all you need," dalam *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, hlm. 1090–1093.
- [13] M. Douze *dkk.*, "The faiss library," *arXiv preprint arXiv:2401.08281*, 2024.
- [14] C. R. Riadi, F. Wulandari, N. Marita, dan D. Nurdwiyanto, "Analisis Deskriptif Perilaku Hate Speech pada Pengguna Sosial Media," *eductum: Jurnal Literasi Pendidikan*, vol. 1, no. 4, hlm. 541–554, 2023.