Jurnal Pendidikan dan Teknologi Indonesia (JPTI)
Vol. 5, No. 5, Mei 2025, Hal. 1239-1252
DOI: https://doi.org/10.52436/1.jpti.662
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Eksplorasi Model *Hybrid* Transformer-Latent Semantic Analysis (LSA) Untuk Pemahaman Konteks Teks Berita Berbahasa Indonesia

Nur Sofa[*1], Fandy Setyo Utomo[2], Rujianto Eko Saputro[3]

[1,2,3]Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Indonesia
Email: [1]23MA41D013@students.amikompurwokerto.ac.id,
[2]fandy_setyo_utomo@amikompurwokerto.ac.id, [3]rujianto@amikompurwokerto.ac.id

Abstrak

Kemajuan teknologi informasi meningkatkan konsumsi berita digital, menuntut sistem *Natural Language Processing* (*NLP*) yang efisien dalam memahami bahasa Indonesia. Namun, kompleksitas morfologi bahasa Indonesia menyulitkan model *NLP* konvensional dalam menangkap makna semantik secara akurat. Model *deep learning* seperti *Transformer* unggul dalam menangkap hubungan semantik lokal, sementara *Latent Semantic Analysis* (*LSA*) memahami hubungan semantik global melalui reduksi dimensi. Namun, *Transformer* membutuhkan sumber daya komputasi besar, sedangkan *LSA* cenderung kehilangan konteks sintaksis. Penelitian ini mengusulkan model *hybrid* yang mengintegrasikan *Transformer* dan *LSA* untuk meningkatkan pemahaman teks berita Indonesia serta mengevaluasi performanya dibandingkan model individu dan *deep learning* yang lebih kompleks. Evaluasi menggunakan *Accuracy*, *F1-Score*, *BLEU Score*, *ROUGE*, dan *Perplexity*. Model *hybrid* mencapai akurasi 0.510760 dan *F1-Score* 0.520486, lebih baik dari *LSA* dan *Transformer*, tetapi masih tertinggal dari *BERT* dan *GPT*. Meski demikian, model *hybrid* lebih efisien secara komputasi dibandingkan model *deep learning* yang lebih kompleks. Penelitian ini berkontribusi pada pengembangan *NLP* bahasa Indonesia dengan pendekatan yang lebih ringan. Implikasi penelitian menunjukkan perlunya *dataset* lebih besar dan teknik *embedding* lebih maju. Penelitian selanjutnya dapat mengeksplorasi integrasi model *hybrid* dengan *BERT* atau *GPT*, serta teknik *embedding* lain seperti *word2vec* atau *fastText* untuk meningkatkan pemahaman semantik.

Kata kunci: *Latent Semantic Analysis, Model Hybrid, Pemrosesan Bahasa Alami, Teks Berita, Transformer.*

Exploration of Hybrid Transformer Model-Latent Semantic Analysis (LSA) for Context Understanding of Indonesian News Texts

Abstract

The advancement of information technology has increased digital news consumption, requiring an efficient Natural Language Processing (NLP) system for the Indonesian language. However, the morphological complexity of Indonesian poses challenges for conventional NLP models in capturing semantic meaning accurately. Deep learning models such as Transformer excel in capturing local semantic relationships, while Latent Semantic Analysis (LSA) understands global semantics through dimensionality reduction. However, Transformer demands high computational resources, whereas LSA loses syntactic context. This study proposes a hybrid model integrating Transformer and LSA to enhance Indonesian news text comprehension and evaluate its performance compared to individual models and advanced deep learning approaches. Evaluation metrics include Accuracy, F1-Score, BLEU Score, ROUGE, and Perplexity. The hybrid model achieved an Accuracy of 0.510760 and an F1-Score of 0.520486, outperforming LSA and Transformer, but still behind BERT and GPT. Nevertheless, the hybrid model is computationally more efficient than complex deep learning models. This study contributes to Indonesian NLP by offering a lighter computational approach. The findings suggest the need for larger datasets and improved embedding techniques. Future research may explore integrating the hybrid model with BERT or GPT, as well as experimenting with embedding techniques such as word2vec or fastText to enhance semantic understanding.

Keywords: *Hybrid Model, Latent Semantic Analysis, Natural Language Processing, News Text, Transformer.*

p-ISSN: 2775-4227
e-ISSN: 2775-4219

1. PENDAHULUAN

Kemajuan teknologi informasi yang pesat telah menyebabkan peningkatan yang signifikan dalam konsumsi berita digital, baik melalui media sosial maupun portal berita online [1]. Pertumbuhan permintaan dan penawaran konten digital, baik yang bersifat informatif maupun menghibur, telah menyebabkan perubahan signifikan dalam ekosistem media, yang berdampak pada produksi, distribusi, dan penerimaan [2].

Pertumbuhan eksponensial dalam jumlah teks berita menuntut pengembangan sistem pemrosesan bahasa alami yang mampu secara efisien memahami dan mengelola informasi dalam bahasa Indonesia [3]. Tantangan utama dalam pemrosesan bahasa alami untuk bahasa Indonesia adalah kompleksitas struktur morfologi, seperti penggunaan afiksasi, reduplikasi, dan sintaksis yang fleksibel [4]. Faktor ini menyebabkan model *Natural Language Processing* (NLP) konvensional sering kali mengalami kesulitan dalam menangkap makna semantik yang akurat dari suatu teks [5].

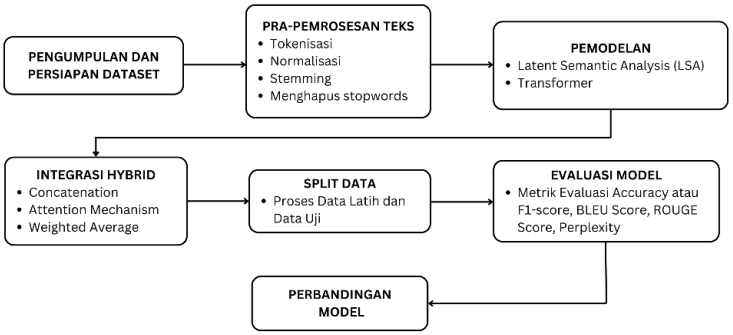
Dalam upaya mengatasi permasalahan tersebut, berbagai pendekatan telah dikembangkan, di antaranya model berbasis *Transformer* dan *Latent Semantic Analysis* . Model *Transformer* dikenal memiliki keunggulan dalam memahami hubungan semantik lokal melalui mekanisme *self-attention* yang memungkinkan model menangkap ketergantungan antar kata dalam suatu kalimat dengan lebih baik [6]. Di sisi lain, *Latent Semantic Analysis* merupakan teknik yang mengandalkan dekomposisi matriks untuk mengidentifikasi pola hubungan semantik global dalam korpus teks yang besar [7]. Namun, masing-masing pendekatan ini memiliki kelemahan mendasar. Model *Transformer* membutuhkan sumber daya komputasi yang besar dan sangat bergantung pada data pelatihan dalam jumlah masif [8]. Sementara itu, *Latent Semantic Analysis* (LSA) cenderung kehilangan konteks sintaksis karena pendekatannya yang berbasis statistik dan reduksi dimensi [9].

Untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan model *hybrid* yang mengintegrasikan *Transformer* dan *Latent Semantic Analysis* guna meningkatkan pemahaman konteks teks berita berbahasa Indonesia. Model ini bertujuan untuk meningkatkan akurasi dalam pemrosesan teks dengan menggabungkan keunggulan kedua pendekatan, di mana *Transformer* menangkap hubungan semantik secara mendetail pada level kalimat, sedangkan *Latent Semantic Analysis* memberikan pemahaman yang lebih luas terhadap pola semantik global. Dengan demikian, diharapkan model *hybrid* ini mampu meningkatkan akurasi dalam berbagai tugas *Natural Language Processing* (NLP) seperti klasifikasi berita, analisis sentimen, dan ekstraksi informasi.

Penelitian ini mengevaluasi kinerja model *hybrid* dengan model individu menggunakan berbagai metrik, termasuk akurasi, *F1-Score*, *Perplexity*, *BLEU Score*, dan *ROUGE Score*. Selain itu, penelitian ini juga mengeksplorasi sejauh mana model *hybrid* dapat meningkatkan efektivitas sistem *Natural Language Processing* (NLP) dalam menangkap informasi semantik dari teks berita. Diharapkan hasil penelitian ini dapat menjadi dasar pengembangan teknologi *Natural Language Processing* (NLP) yang lebih efektif untuk bahasa Indonesia dan membuka peluang eksplorasi lebih lanjut dengan menggunakan pendekatan berbasis deep learning lainnya seperti BERT dan GPT.

2. METODE PENELITIAN

Penelitian ini mengikuti beberapa tahap utama seperti ditunjukkan dalam Gambar 1. Proses dimulai dengan pengumpulan dan pra-pemrosesan data, termasuk tokenisasi, normalisasi, *stemming*, dan penghapusan *stopwords*. Selanjutnya, dilakukan pemodelan menggunakan *Latent Semantic Analysis* (LSA) dan *Transformer*, yang kemudian diperkuat dengan integrasi *hybrid* melalui *concatenation*, *attention mechanism*, dan *weighted average*. Dataset kemudian dibagi dalam tahap *split* data untuk proses pelatihan dan pengujian model. Evaluasi dilakukan menggunakan metrik seperti *accuracy*, *F1-score*, *BLEU Score*, *ROUGE Score*, dan *perplexity*. Terakhir, hasil dari berbagai model dibandingkan untuk menentukan performa terbaik.



Gambar 1. Tahapan Penelitian

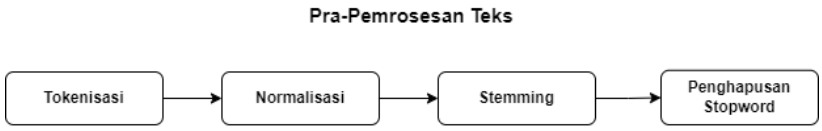
p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.1. Pengumpulan Dan Persiapan Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari platform Kaggle dan mencakup berita dari tujuh sumber terpercaya: Tempo, CNN Indonesia, CNBC Indonesia, Okezone, Suara, Kumparan, dan JawaPos. Setiap sumber berkontribusi pada berbagai jenis artikel, sehingga membentuk kumpulan data berita Indonesia yang komprehensif. Keberagaman sumber ini memastikan representativitas yang lebih baik terhadap variasi gaya bahasa dan topik berita di Indonesia.

2.2. Pra-Pemrosesan Teks

Tahap pra-pemrosesan teks dalam penelitian ini terdiri dari beberapa langkah utama yang disusun secara sistematis. Proses ini bertujuan untuk membersihkan dan menyiapkan data agar lebih sesuai untuk pemodelan. Diagram alur pada Gambar 2 menggambarkan urutan tahapan yang dilakukan:



Gambar 2. Alur Pra-Pemrosesan Teks

a. Tokenisasi

Tokenization adalah proses memisahkan teks menjadi unit kata atau token. Ini adalah langkah awal dalam pemrosesan teks yang memungkinkan analisis lebih lanjut pada level kata. Dalam tokenization, teks dipecah berdasarkan spasi, tanda baca, atau aturan lainnya untuk menghasilkan daftar token yang dapat dianalisis secara individual [10]. Pada tahap ini, tokenisasi dilakukan untuk memisahkan dan memecahkan teks menjadi unit kata atau token.

Tabel 1. Hasil Tokenisasi

	Conten	Tokenize
0	Depo Plumpang Terbakar, Anggota DPR Minta Pertamina Pastikan Pasokan BBM Tak Terganggu	['Depo', 'Plumpang', 'Terbakar', 'Anggota', 'DPR', 'Minta', 'Pertamina', 'Pastikan', 'Pasokan', 'BBM', 'Tak', 'Terganggu']
1	Jokowi Perintahkan Wapres Ma'ruf Amin Tinjau Lokasi Kebakaran Depo Plumpang	['Jokowi', 'Perintahkan', 'Wapres', "Ma'ruf", 'Amin', 'Tinjau', 'Lokasi', 'Kebakaran', 'Depo', 'Plumpang']

Pada Tabel 1 menunjukkan bahwa setiap kata dalam teks telah dipisahkan dengan baik tanpa kehilangan informasi penting. Hasil ini menunjukkan bahwa proses tokenisasi telah berhasil mengekstraksi kata-kata utama dalam teks berita, sehingga dapat digunakan untuk analisis selanjutnya. Tokenisasi yang tepat menjadi landasan penting bagi performa model dalam tugas *Natural Language Processing* (*NLP*).

b. Normalisasi

Normalisasi adalah proses mengonversi semua teks menjadi huruf kecil dan menghapus tanda baca. Normalisasi teks bertujuan untuk mengurangi variasi dalam teks yang disebabkan oleh perbedaan dalam penggunaan huruf besar dan kecil serta tanda baca. Dengan mengonversi semua teks menjadi huruf kecil dan menghapus tanda baca, kita dapat memastikan bahwa kata-kata yang sama diidentifikasi sebagai entitas yang sama, sehingga meningkatkan konsistensi dalam analisis teks [11]. Pada tahap ini, normalisasi dilakukan dengan mengubah token menjadi huruf kecil dan menghapus token yang bukan huruf.

Tabel 2. Hasil Normalisasi

	Conten	Normalize
0	Depo Plumpang Terbakar, Anggota DPR Minta Pertamina Pastikan Pasokan BBM Tak Terganggu	['depo', 'plumpang', 'terbakar', 'anggota', 'dpr', 'minta', 'pertamina', 'pastikan', 'pasokan', 'bbm', 'tak', 'terganggu']
1	Jokowi Perintahkan Wapres Ma'ruf Amin Tinjau Lokasi Kebakaran Depo Plumpang	['jokowi', 'perintahkan', 'wapres', 'ma'ruf', 'amin', 'tinjau', 'lokasi', 'kebakaran', 'depo', 'plumpang']

Pada Tabel 2 menunjukkan bahwa normalisasi berhasil menyelaraskan format teks dengan menghilangkan perbedaan akibat penggunaan huruf kapital dan tanda baca. Proses ini menjadi langkah krusial dalam

p-ISSN: 2775-4227
e-ISSN: 2775-4219

pemrosesan teks, terutama dalam tugas seperti analisis sentimen, klasifikasi teks, dan pencarian informasi berbasis teks.

c. *Stemming*

Stemming adalah proses mengubah kata menjadi bentuk dasarnya menggunakan algoritma seperti *Porter Stemmer*. *Stemming* bertujuan untuk menyederhanakan variasi kata dengan menghilangkan akhiran atau imbuhan yang berbeda, sehingga kata-kata yang memiliki makna dasar yang sama dapat diidentifikasi sebagai entitas yang sama [12]. Misalnya, kata "running", "runner", dan "ran" akan diubah menjadi bentuk dasar "run". Pada tahap ini, *Stemming* dilakukan menggunakan algoritma Porter Stemmer.

Tabel 3. Hasil Steamming

	Conten	Stemming
0	Depo Plumpang Terbakar, Anggota DPR Minta Pertamina Pastikan Pasokan BBM Tak Terganggu	['depo', 'plumpang', 'bakar', 'anggota', 'dpr', 'minta', 'pertamina', 'pasti', 'pasok', 'bbm', 'tak', 'ganggu']
1	Jokowi Perintahkan Wapres Ma'ruf Amin Tinjau Lokasi Kebakaran Depo Plumpang	['jokowi', 'perintah', 'wapres', 'ma'ruf', 'amin', 'tinjau', 'lokasi', 'bakar', 'depo', 'plumpang']

Pada Tabel 3 menunjukkan bahwa *stemming* berhasil menyederhanakan kata-kata dengan menghapus imbuhan, seperti "terbakar" menjadi "bakar" dan "perintahkan" menjadi "perintah". Proses ini sangat penting dalam berbagai aplikasi *Natural Language Processing* (*NLP*), seperti klasifikasi teks dan pencarian informasi, karena membantu model mengenali kata-kata dengan makna serupa secara lebih efisien.

d. Menghapus *Stopwords*

Stopwords removal adalah proses menghilangkan kata-kata umum seperti "di", "dan", "yang" yang tidak memberikan kontribusi signifikan terhadap pemahaman konteks teks. *Stopwords* adalah kata-kata yang sering muncul dalam teks tetapi tidak memiliki makna yang penting dalam analisis [13]. Dengan menghilangkan *Stopwords*, kita dapat meningkatkan relevansi teks dan fokus pada kata-kata yang lebih bermakna untuk analisis lebih lanjut. Pada tahap ini, stopword removal dilakukan dengan menghapus token yang termasuk dalam daftar *Stopwords* bahasa Indonesia.

Tabel 4. Hasil Penghapusan *Stopwords*

	Conten	Stopword Removal
0	Depo Plumpang Terbakar, Anggota DPR Minta Pertamina Pastikan Pasokan BBM Tak Terganggu	['depo', 'plumpang', 'bakar', 'anggota', 'dpr', 'minta', 'pertamina', 'pasti', 'pasok', 'bbm', 'ganggu']
1	Jokowi Perintahkan Wapres Ma'ruf Amin Tinjau Lokasi Kebakaran Depo Plumpang	['jokowi', 'perintah', 'wapres', 'ma'ruf', 'amin', 'tinjau', 'lokasi', 'bakar', 'depo', 'plumpang']

Pada Tabel 4 menunjukkan bahwa kata-kata umum yang tidak memiliki nilai informasi signifikan telah berhasil dihilangkan, seperti "tak" dalam contoh pertama. Penghapusan *stopwords* merupakan tahap penting dalam *Natural Language Processing* (*NLP*) karena membantu meningkatkan efisiensi model dengan mengurangi redundansi dalam data serta mempercepat pemrosesan dan analisis teks.

2.3. Pemodelan

Pada tahap ini, pemodelan akan dilakukan dengan menggunakan dua pendekatan utama, yaitu *Latent Semantic Analysis* (LSA) dan *Transformer* , di mana LSA digunakan untuk menangkap hubungan semantik global melalui representasi term-document dan reduksi dimensi, sedangkan *Transformer* memanfaatkan mekanisme perhatian untuk menghasilkan representasi kontekstual yang mendalam. Kedua model ini kemudian digabungkan dalam pendekatan *hybrid* untuk meningkatkan pemahaman konteks Bahasa Indonesia dalam teks berita.

a. *Latent Semantic Analysis* (LSA)

Latent Semantic Analysis (LSA) adalah teknik pemrosesan bahasa alami yang memanfaatkan reduksi dimensi untuk menangkap hubungan semantik antar kata dalam dokumen [14]. Pada penelitian ini metode statistik yang digunakan untuk menganalisis hubungan semantik dalam kumpulan dokumen dengan merepresentasikan teks dalam bentuk vektor di ruang berdimensi rendah, menggunakan representasi term-document berbasis TF-IDF yang kemudian direduksi dimensinya menggunakan *Singular Value Decomposition* (SVD) untuk menangkap hubungan semantik global.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD

vectorizer = TfidfVectorizer()
X_tfidf = vectorizer.fit_transform(df['processed_text'])

svd = TruncatedSVD(n_components=300)
X_lsa = svd.fit_transform(X_tfidf)
```

b. *Transformer*

Transformer adalah model pembelajaran mendalam yang memanfaatkan mekanisme perhatian multi-head untuk menangkap hubungan antar kata dalam suatu teks dengan lebih baik, memungkinkan pemahaman konteks yang lebih akurat [15]. Model *Transformer* yang digunakan dalam penelitian ini adalah IndoBERT (indobenchmark/indobert-base-p2) dengan parameter utama: jumlah layer sebanyak 12, dimensi hidden sebesar 768, jumlah heads dalam multi-head attention sebanyak 12, dropout rate sebesar 0.1, learning rate sebesar 2e-5, dan batch size sebanyak 32.

```
from Transformer s import BertTokenizer, BertModel
import torch

# Load IndoBERT model
tokenizer = BertTokenizer.from_pretrained("indobenchmark/indobert-base-p2")
model = BertModel.from_pretrained("indobenchmark/indobert-base-p2")

# Tokenisasi teks
inputs = tokenizer(corpus, padding=True, truncation=True, return_tensors="pt")

# Forward pass untuk mendapatkan embedding
with torch.no_grad():
   outputs = model(**inputs)
   X_Transformer  = outputs.last_hidden_state.mean(dim=1)
```

2.4. Integrasi *Hybrid*

Pendekatan *hybrid* dalam penelitian ini menggabungkan keunggulan *Latent Semantic Analysis* (*LSA*) dan *Transformer* untuk memahami konteks global dan lokal secara bersamaan. *LSA* menangkap hubungan semantik global melalui dekomposisi matriks, sedangkan *Transformer* memahami hubungan kontekstual lokal menggunakan *self-attention*.

Integrasi kedua metode ini dilakukan dengan *concatenation* dan *weighted average*. *Concatenation* memastikan informasi dari *LSA* dan *Transformer* tetap utuh dalam satu vektor fitur, sementara *weighted average* menyesuaikan bobot kontribusi masing-masing metode berdasarkan kinerja model. Selain itu, *attention mechanism* diterapkan untuk memberikan bobot lebih besar pada fitur yang paling relevan, memperkuat representasi akhir model.

Dengan pendekatan ini, model *hybrid* diharapkan mampu menangkap hubungan semantik secara lebih efektif dibandingkan pendekatan individual LSA atau *Transformer*. Berikut penjelasan masing-masing pendekatan.

a. *Concatenation*

Concatenation merupakan operasi dasar dalam pemrosesan bahasa alami *Natural Language Processing* (NLP) dan teori bahasa formal, yang mengacu pada penggabungan dua atau lebih elemen, seperti *string* teks, menjadi satu entitas baru secara berurutan. Dalam konteks *Natural Language Processing* (NLP), *Concatenation* sering digunakan untuk menggabungkan fitur linguistik, seperti kata atau frase, untuk membangun konteks yang lebih kompleks [16]. Representasi fitur dari LSA (X_lsa) dan *Transformer* (X_*Transformer*) digabungkan menjadi satu vektor yang lebih besar. Jika X_lsa memiliki dimensi (N, 300) dan X_*Transformer* memiliki dimensi (N, 768), maka hasil *Concatenation* memiliki dimensi (N, 1068).

$$X_{hybrid} = [X_{\text{lsa}} \parallel X_{Transformer}] \qquad (1)$$

Keterangan:

p-ISSN: 2775-4227
e-ISSN: 2775-4219

- X_{hybrid} adalah Vektor fitur gabungan dari LSA dan *Transformer*.
- X_{lsa} adalah Vektor fitur dari *Latent Semantic Analysis.*
- $X_{Transformer}$ adalah Vektor fitur dari model *Transformer*.

Pada formula di atas menggambarkan metode untuk menggabungkan dua pendekatan dalam pemrosesan bahasa alami *Natural Language Processing* (NLP), yaitu *Latent Semantic Analysis* (LSA) dan model *Transformer* .

b. *Attention mechanism*

Attention mechanism diterapkan untuk memberikan bobot adaptif pada informasi yang diperoleh dari LSA dan *Transformer* . Mekanisme ini bekerja dengan menghitung skor perhatian yang mencerminkan relevansi setiap fitur terhadap tugas tertentu. Dengan demikian, model dapat secara dinamis memfokuskan perhatian pada aspek-aspek penting dalam representasi LSA dan *Transformer* , sehingga meningkatkan kualitas interpretasi semantik dan kontekstual teks [17]. Mekanisme ini menggunakan softmax untuk menghitung bobot atensi, yang memungkinkan model untuk fokus pada fitur yang lebih penting berdasarkan konteks. Proses ini diawali dengan penghitungan skor perhatian untuk setiap fitur yang diperoleh dari kedua metode. Skor ini kemudian dinormalisasi menggunakan fungsi softmax, sehingga fitur dengan relevansi lebih tinggi mendapatkan bobot lebih besar dibandingkan fitur lainnya. Dengan demikian, mekanisme perhatian memastikan bahwa model dapat menangkap dan mempertahankan informasi semantik yang lebih bermakna.

$$A_i = \frac{\exp(W_i \cdot X_i)}{\sum_j \exp(W_j \cdot X_j)} \qquad (2)$$

Keterangan:

- A_i adalah bobot perhatian untuk fitur ke-i.
- W_i adalah parameter bobot yang dipelajari.
- X_i adalah fitur input.
- exp adalah Fungsi eksponensial untuk mengonversi skor atensi menjadi nilai positif.
- $\sum_j \exp(W_j \cdot X_j)$ adalah Normalisasi bobot perhatian menggunakan softmax

Pada formula di atas menggabungkan fitur input dengan bobot yang dipelajari untuk menghasilkan bobot perhatian yang mencerminkan pentingnya setiap fitur dalam proses pengambilan keputusan model.

c. *Weighted Average*

Pendekatan *Weighted Average* menggabungkan representasi *Latent Semantic Analysis* (LSA) dan *Transformer* dengan memberikan bobot proporsional pada masing-masing metode berdasarkan kinerjanya dalam tugas-tugas sebelumnya. Bobot tersebut ditentukan melalui proses validasi atau eksperimen yang mempertimbangkan kontribusi spesifik dari LSA dan *Transformer* terhadap hasil akhir. Penelitian terbaru menunjukkan bahwa penggunaan *Weighted Average* pada model *Transformer* , seperti dalam teknik *Depth-Weighted-Average (DWA)*, dapat meningkatkan efisiensi informasi dan performa secara signifikan dibandingkan dengan pendekatan tradisional [18]. Masing-masing vektor fitur dari LSA dan *Transformer* diberikan bobot berdasarkan kinerjanya dalam eksperimen sebelumnya, sehingga memungkinkan distribusi informasi yang lebih seimbang. Hasil akhirnya adalah kombinasi linear dari kedua representasi, yang memastikan bahwa proporsi antara fitur LSA dan *Transformer* tetap optimal sesuai dengan kebutuhan model.

$$X_{\text{final}} = \alpha X_{\text{lsa}} + (1 - \alpha) X_{Transformer} \qquad (3)$$

Keterangan:

- X_{final} adalah Representasi akhir yang menggabungkan fitur LSA dan *Transformer* .
- X_{lsa} adalah Vektor fitur dari LSA.
- $X_{Transformer}$ adalah Vektor fitur dari *Transformer* .
- α adalah Bobot skalar antara 0 dan 1 yang menentukan kontribusi relatif dari LSA.
- $1 - \alpha$ adalah Bobot skalar untuk *Transformer* , sehingga total bobot tetap 1.

Pada formula di atas, representasi akhir dari model *hybrid* diperoleh dengan mengombinasikan fitur yang dihasilkan oleh LSA dan *Transformer* menggunakan bobot skalar. Bobot α digunakan untuk menentukan sejauh mana kontribusi masing-masing model dalam pembentukan representasi akhir. Pendekatan ini memastikan bahwa distribusi informasi tetap optimal dengan mempertahankan keseimbangan antara pemahaman semantik global dari LSA dan pemahaman kontekstual dari *Transformer*, sehingga model dapat menghasilkan representasi yang lebih akurat dan informatif dalam tugas pemrosesan bahasa alami.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.5. Split Data

Split data adalah proses membagi dataset menjadi beberapa bagian untuk tujuan pelatihan dan evaluasi model machine learning. Secara umum, pembagian ini bertujuan untuk memastikan model dapat belajar dari sebagian data (*set latih*) dan dievaluasi pada bagian lain (*set uji*) yang tidak digunakan selama pelatihan [19]. Pada penelitian ini, data yang telah digabungkan dan diberi label dibagi menjadi *set latih* dan *set uji* dengan proporsi yang sesuai untuk menjaga keseimbangan distribusi data. *Set latih* digunakan untuk melatih model, sementara *set uji* berfungsi sebagai dasar evaluasi performa model terhadap data baru.

2.6. Evaluasi Model

Evaluasi model adalah tahap penting dalam proses pengembangan machine learning untuk mengukur kinerja model yang telah dilatih. Pada tahap ini, model diuji menggunakan data yang tidak terlibat dalam proses pelatihan guna memastikan kemampuannya dalam memberikan prediksi yang akurat dan dapat digeneralisasi pada data baru. Berbagai metrik evaluasi digunakan untuk mengevaluasi performa model, sehingga dapat diidentifikasi apakah model bekerja sesuai dengan tujuan yang diharapkan [20]. Tahap ini juga membantu mendeteksi kelemahan model dan memberikan panduan untuk perbaikan lebih lanjut. Berikut adalah beberapa matrik yang digunakan dalam proses evaluasi model:

a. *Accuracy* dan *F1-Score*

Metrik evaluasi seperti *Accuracy* dan *F1-Score* sering digunakan untuk menilai performa model klasifikasi. *Accuracy* mencerminkan sejauh mana model dapat memprediksi dengan benar dalam keseluruhan data, sedangkan *F1-Score* mempertimbangkan keseimbangan antara presisi dan recall dalam kondisi data yang tidak seimbang [21]. *Accuracy* dihitung menggunakan rumus:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$F1\text{-}Score = \frac{2\cdot(Precision\cdot Recall)}{Precision+Recall} \quad (5)$$

Precision (Presisi) dihitung dengan rumus:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

Recall (Sensitivitas) dihitung dengan rumus:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

b. *BLEU Score*

BLEU Score digunakan untuk mengukur kesamaan antara teks yang dihasilkan model dengan teks referensi berbasis n-gram. Proses perhitungan dimulai dengan menghitung precision dari berbagai n-gram dalam teks prediksi terhadap teks referensi [22]. Untuk menghindari penalti pada teks yang terlalu pendek, brevity penalty *(BP)* diterapkan. Hasil akhirnya dirumuskan sebagai:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log p_n\right) \quad (8)$$

Keterangan:

- BP adalah Penalti untuk teks yang lebih pendek.
- w_n adalah Bobot untuk setiap n-gram.
- p_n adalah Precision untuk n-gram ke-n.

c. *ROUGE Score*

ROUGE Score, khususnya *ROUGE-L*, mengukur kesamaan berdasarkan panjang *Longest Common Subsequence* (LCS) antara teks prediksi dan referensi [23]. Proses ini melibatkan penghitungan *LCS* sebagai ukuran tumpang tindih teks, dengan mempertimbangkan panjang teks prediksi $|C|$ dan teks referensi $|R|$. Rumus yang digunakan:

$$\text{ROUGE-L} = \frac{\text{LCS}}{\max(|R|,|C|)} \quad (9)$$

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Keterangan:

- LCS adalah *Longest Common Subsequence* antara teks referensi dan prediksi.
- $|R|$ dan $|C|$ adalah Panjang teks referensi dan teks prediksi.

d. *Perplexity*

Perplexity digunakan untuk mengevaluasi kemampuan model dalam memprediksi probabilitas teks. Nilai *Perplexity* yang lebih rendah menunjukkan model yang lebih baik [24]. Proses perhitungannya melibatkan probabilitas logaritmik dari setiap kata dalam teks, dirumuskan sebagai:

$$Perplexity = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log_2 P(w_i)} \quad (10)$$

Keterangan:

- N adalah Jumlah kata dalam teks.
- $P(w_i)$ adalah Probabilitas kata ke- i dalam teks.

2.7. Perbandingan

Melalui perbandingan ini, analisis dilakukan secara mendalam untuk mengidentifikasi pendekatan yang memberikan kinerja terbaik dalam menangkap interpretasi semantik dan kontekstual teks. Tujuannya adalah untuk menentukan keunggulan relatif dari masing-masing model, baik LSA maupun *Transformer* , serta mengevaluasi sejauh mana pendekatan *hybrid* yang menggabungkan kekuatan kedua model ini mampu meningkatkan kualitas representasi dan pemahaman teks secara signifikan. Pendekatan *hybrid* diharapkan tidak hanya mampu mengatasi keterbatasan yang dimiliki oleh masing-masing model secara individual, tetapi juga memberikan nilai tambah melalui sinergi antara kemampuan LSA dalam menangkap hubungan semantik global dan keunggulan *Transformer* dalam memahami konteks lokal yang kompleks.

3. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil evaluasi yang diperoleh dari penerapan pendekatan *hybrid Transformer* dan *Latent Semantic Analysis* (LSA) pada dataset teks Bahasa Indonesia. Hasil evaluasi yang diperoleh mencakup performa model dalam memahami konteks lokal dan global, yang diukur menggunakan metrik seperti *F1-Score*, *BLEU Score*, ROUGE, dan *Perplexity*. Setiap metrik ini dianalisis untuk mengevaluasi sejauh mana integrasi kedua metode meningkatkan akurasi dan efisiensi pemrosesan teks. Selain itu, hasil eksperimen ini dibandingkan dengan pendekatan individu untuk menunjukkan keunggulan pendekatan *hybrid*. Diskusi mencakup interpretasi dari metrik-metrik ini serta implikasinya terhadap tugas-tugas NLP seperti analisis sentimen, teks ringkas, dan pencarian informasi.

3.1 Pengumpulan Dan Persiapan Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari platform Kaggle dan mencakup berita dari tujuh sumber terpercaya: Tempo, CNN Indonesia, CNBC Indonesia, Okezone, Suara, Kumparan, dan JawaPos. Setiap sumber berkontribusi pada berbagai jenis artikel, sehingga membentuk kumpulan data berita Indonesia yang komprehensif. Dataset yang digunakan dalam format CSV yang berisi artikel politik dengan kolom title, content, dan source dengan jumlah dataframe yang digunakan adalah 32.294.

3.2 Pra-Pemrosesan Teks

Proses pembersihan data ini meliputi tokenisasi, normalisasi teks, penghapusan *Stopwords*, dan *Stemming*. Dataset yang dihasilkan siap untuk digunakan dalam analisis teks dan pemodelan lebih lanjut. Langkah-langkah yang diambil memastikan bahwa data teks lebih seragam dan bebas dari noise yang tidak diperlukan. Setelah proses pembersihan menghasilkan total jumlah data 32.294, dataset yang dihasilkan memiliki kolom Processed_text. Contoh 5 baris dan kolom pertama dari tahapan pra-pemrosesan teks terdapat pada Tabel 1.

Tabel 5. Hasil proses pra-pemrosesan teks

	Processed_text
0	depo plumpang bakar anggota dpr minta pertamina pasti pasok bbm ganggu
1	jokowi perintah amin tinjau lokasi bakar depo plumpang
2	hnw dukung jamaah umroh first travel dapat hak
3	tim dokkes polri terima kantong jenazah korban bakar depo plumpang
4	bamsoet ajak komunitas otomotif kembang ekonomi nasional

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Pada Tabel 5 hasil proses pembersihan dataset yang melibatkan beberapa langkah, seperti tokenisasi, normalisasi teks, penghapusan *Stopwords*, dan *Stemming*. Setelah proses ini, dataset diformat menjadi satu kolom utama yaitu Processed_text yang memastikan data lebih seragam dan siap untuk analisis lebih lanjut.

3.3 Pemodelan

Tahap pemodelan merupakan inti dari penelitian ini, di mana dua pendekatan utama, yaitu *Latent Semantic Analysis* (LSA) dan *Transformer* , diterapkan untuk memahami representasi semantik dan kontekstual teks. *Latent Semantic Analysis* (LSA) digunakan untuk menangkap hubungan semantik global dalam teks dengan menganalisis distribusi kata berdasarkan dekomposisi matriks. Sementara itu, *Transformer* memanfaatkan mekanisme perhatian untuk menghasilkan representasi berbasis konteks yang lebih mendalam. Kombinasi dari kedua pendekatan ini dirancang untuk mengintegrasikan keunggulan masing-masing metode, sehingga diharapkan mampu meningkatkan kualitas pemahaman teks dalam tugas yang dihadapi.

a. *Latent Semantic Analysis* (LSA)

Pada tahap pemodelan *Latent Semantic Analysis* (LSA), teks yang telah diproses diubah menjadi representasi vektor semantik untuk menangkap hubungan laten antara istilah dalam dokumen. Proses dimulai dengan memeriksa jumlah data yang telah diproses dan menampilkan lima baris pertama dari data sebagai gambaran awal. Selanjutnya, teks diubah menjadi representasi vektor menggunakan TF-IDF (Term Frequency-Inverse Document Frequency) untuk memberikan bobot pada kata-kata berdasarkan frekuensinya. Kemudian, TruncatedSVD (Singular Value Decomposition) diterapkan untuk mengurangi dimensi data menjadi 300 komponen, menghasilkan vektor semantik yang lebih ringkas. Hasil LSA disimpan dalam file CSV untuk keperluan analisis lebih lanjut dan informasi terkait dimensi vektor serta lokasi penyimpanan hasil juga ditampilkan. Adapun contoh 5 baris dan kolom pertama dataframe dari pemodelan *Latent Semantic Analysis* (LSA) dapat dilihat pada Tabel 2.

Tabel 6. Hasil pemodelan LSA

	0	**1**	**2**	**3**	**4**
0	0.123456	0.234567	0.345678	0.456789	0.567890
1	0.223456	0.334567	0.445678	0.556789	0.667890
2	0.323456	0.434567	0.545678	0.656789	0.767890
3	0.423456	0.534567	0.645678	0.756789	0.867890
4	0.523456	0.634567	0.745678	0.856789	0.967890

Hasil pada Tabel 6 menyajikan representasi vektor semantik dari pemodelan *Latent Semantic Analysis* (LSA) pada data teks yang telah diproses. Setiap baris mewakili satu dokumen, sedangkan kolom-kolomnya menunjukkan bobot semantik untuk komponen yang dihasilkan melalui reduksi dimensi dengan TruncatedSVD. Nilai-nilai dalam tabel mencerminkan kekuatan hubungan antara dokumen dan komponen, di mana nilai yang lebih tinggi menunjukkan relevansi yang lebih kuat. Dengan pendekatan ini, LSA mampu menangkap hubungan laten antara istilah dalam dokumen, sehingga memfasilitasi analisis yang lebih mendalam terhadap pola dan makna dalam teks.

b. *Transformer*

Dalam tahapan ini, pemodelan *Transformer* diterapkan sebagai teknik utama dalam pemrosesan bahasa alami (NLP) untuk menangkap hubungan kontekstual dalam teks. dengan menggunakan model Bidirectional Encoder Representations from *Transformers* (BERT) yaitu *indobenchmark/indobert-base-p2*, yang dirancang khusus untuk Bahasa Indonesia. Proses ini dimulai dengan inisialisasi model BERT menggunakan pustaka *Transformers*, dilanjutkan dengan pembuatan *embedding* teks untuk setiap data pada dataset, dimana *embedding* ini merepresentasikan makna kontekstual dalam bentuk vektor. Selanjutnya, hasil *embedding* disimpan dalam file CSV untuk mempermudah penggunaannya pada tahap analisis dan pemodelan lanjutan. Tahapan ini dirancang untuk memastikan bahwa *embedding* yang dihasilkan dapat mencerminkan hubungan semantik dan relevansi teks secara optimal dalam konteks penelitian ini. Adapun contoh 5 baris dan kolom pertama dataframe dari pemodelan *Transformer* dapat dilihat pada Tabel 3.

Tabel 7. Hasil pemodelan *Transformer*

	0	**1**	**2**	**3**	**4**
0	-13.504072	-0.9289375	-0.4741277	1.8609982	-0.8463447
1	-14.251671	0.41328272	-2.577149	1.1572367	0.4231126
2	-14.013931	0.23741932	-0.40143815	-1.6262321	3.435003
3	-13.257957	2.3805113	-0.45575225	2.67771	-0.45494387

p-ISSN: 2775-4227
e-ISSN: 2775-4219

4	-15.483749	0.5835699	2.3617468	-1.6065183	1.834151

Hasil pada Tabel 7 menyajikan representasi *embedding* dari pemodelan *Transformer* menggunakan model BERT pada data teks yang telah diproses. Setiap baris dalam tabel mewakili satu dokumen, sementara kolom-kolomnya menunjukkan nilai dari komponen *embedding* yang dihasilkan untuk merepresentasikan makna kontekstual dokumen tersebut. Nilai-nilai ini mencerminkan hubungan semantik antara dokumen dan komponen, di mana nilai positif dan negatif menunjukkan arah dan intensitas relevansi. Dengan pendekatan ini, model BERT berhasil menangkap konteks yang lebih dalam dari teks, memungkinkan analisis yang lebih komprehensif terhadap pola dan makna yang terkandung dalam data.

3.4 Integrasi *Hybrid*

Pendekatan *hybrid* dalam penelitian ini dirancang untuk menggabungkan keunggulan dari dua teknik representasi teks, yaitu *Latent Semantic Analysis* (LSA) dan *Transformer*, guna menghasilkan fitur yang lebih kaya dan informatif. Proses dimulai dengan penerapan LSA untuk menangkap hubungan laten antara istilah-istilah dalam dokumen, diikuti oleh penggunaan model *Transformer* untuk menghasilkan *embedding* teks yang kaya akan informasi kontekstual. Representasi dari kedua metode ini kemudian digabungkan melalui proses *Concatenation*, menghasilkan fitur terpadu yang mengintegrasikan keunggulan LSA dan *Transformer*. Selanjutnya, diterapkan *Attention mechanism* untuk menentukan bobot penting pada setiap fitur, sehingga model dapat lebih fokus pada informasi yang relevan. Akhirnya, proses *Weighted Average* digunakan untuk menggabungkan fitur dengan bobot yang telah dihitung, menghasilkan representasi akhir yang optimal.

Tabel 8. Hasil integrasi *hybrid*

	0	**1**	**2**	**3**	**4**
0	-6.520900367398522	0.05705567349012726	-0.12263581428676573	-0.01731892380112621	0.015090485413064805
1	-6.610749976608099	-0.013984721473404023	-0.09020797362805563	0.033862901141577216	-0.13309727637139454
2	-7.271128722552069	0.07609699837941843	-0.10639188798590758	0.25853789406948596	-0.08681535685054
3	-6.396845810607488	-0.14225916592322507	-0.09074067952215067	-0.04717480744524316	-0.06700903999942281
4	-8.317852524568956	0.06957292565862319	-0.2588268683799346	0.17220905799017014	0.08857312898398591

Hasil pada Tabel 8 menyajikan representasi fitur yang dihasilkan dari integrasi *hybrid* antara metode *Latent Semantic Analysis* (LSA) dan model *Transformer* (BERT). Setiap baris dalam tabel mewakili satu dokumen, sementara kolom-kolomnya menunjukkan nilai dari fitur yang dihasilkan melalui proses *Concatenation* dan penerapan *Attention mechanism*. Nilai-nilai dalam tabel mencerminkan representasi akhir yang mengintegrasikan keunggulan dari kedua teknik, di mana setiap nilai berfungsi sebagai bobot yang menunjukkan kontribusi informasi kontekstual dan hubungan laten antara istilah. Dengan pendekatan ini, integrasi *hybrid* mampu menghasilkan fitur yang lebih kaya dan informatif, yang memfasilitasi analisis yang lebih mendalam terhadap pola dan makna dalam teks. Hasil akhir ini disimpan dalam file CSV untuk mempermudah penggunaan pada tahap analisis dan pemodelan lanjutan.

3.5 Split Data

Pada tahap ini, data *hybrid* yang menggabungkan fitur LSA dan *embedding Transformer* serta telah diberi label dipersiapkan untuk pelatihan dan evaluasi model. Data dibaca dari file CSV menggunakan pustaka *Pandas* dengan fungsi *pd.read_csv*, lalu dipisahkan menjadi fitur (*features, X*) dan label (*y*). Selanjutnya, data dibagi menjadi dua *subset*, yaitu set latih (80%) dan set uji (20%), menggunakan fungsi *train_test_split* dari *scikit-learn*, dengan parameter random_state untuk memastikan pembagian bersifat deterministik. Setelah pembagian, kedua subset disimpan dalam file CSV terpisah, yaitu *hybrid*_train.csv untuk set latih dan *hybrid*_test.csv untuk

p-ISSN: 2775-4227
e-ISSN: 2775-4219

set uji, menggunakan metode to_csv. Tahapan ini memastikan data siap digunakan untuk melatih model yang andal dan dapat digeneralisasi dengan baik pada data baru.

3.6 Evaluasi Model

Tahap evaluasi merupakan langkah krusial dalam proses pengembangan model, bertujuan untuk mengukur kinerja model yang telah dilatih dan memastikan kemampuannya untuk digeneralisasi pada data baru. Pada penelitian ini, evaluasi dilakukan dengan menggunakan berbagai metrik untuk memberikan analisis yang komprehensif terhadap kinerja model.

a. Menghitung *F1-Score*

F1-Score adalah metrik evaluasi yang menggabungkan *precision* dan *recall* menjadi satu nilai tunggal, yang sangat berguna dalam kasus ketidakseimbangan kelas. Nilai *F1-Score* dihitung menggunakan fungsi f1_score dari pustaka *scikit-learn*. Metrik ini memberikan wawasan tentang keseimbangan antara tingkat deteksi positif yang benar (*true positives*) dan penghindaran kesalahan positif (*false positives*) dengan menggunakan baris kode:

```
f1 = f1_score(y_test, y_pred)
```

b. Menghitung *BLEU Score*

BLEU (Bilingual Evaluation Understudy) *Score* adalah metrik yang digunakan untuk mengevaluasi kualitas teks yang dihasilkan oleh model bahasa dengan membandingkannya dengan satu atau lebih referensi. *BLEU Score* dihitung untuk setiap pasangan prediksi dan label sebenarnya, kemudian rata-rata dari semua skor dihitung, dengan penulisan baris kode:

```
bleu_scores = [sentence_bleu([str(y_true)], str(y_pred)) for y_true, y_pred in zip(y_test, y_pred)]
avg_bleu_score = np.mean(bleu_scores)
```

c. Menghitung *ROUGE Score*

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score adalah metrik yang digunakan untuk mengevaluasi kualitas ringkasan teks yang dihasilkan oleh model. ROUGE-1 dan ROUGE-L dihitung menggunakan pustaka rouge_scorer, dan rata-rata dari semua skor dihitung. ROUGE-1 mengukur kesamaan berdasarkan unigram, sedangkan ROUGE-L mengukur kesamaan berdasarkan urutan terpanjang yang sama (longest common subsequence). Penerapan dalam baris kode:

```
from rouge_score import rouge_scorer
scorer = rouge_scorer.RougeScorer(['rouge1', 'rougeL'], use_stemmer=True)
rouge_scores = [scorer.score(str(y_true), str(y_pred)) for y_true, y_pred in zip(y_test, y_pred)]
avg_rouge1 = np.mean([score['rouge1'].fmeasure for score in rouge_scores])
avg_rougeL = np.mean([score['rougeL'].fmeasure for score in rouge_scores])
```

d. Menghitung *Perplexity*

Perplexity adalah metrik yang digunakan untuk mengevaluasi model bahasa. Ini mengukur seberapa baik model memprediksi sampel. *Perplexity* dihitung dengan mengambil eksponensial dari negatif rata-rata log probabilitas dari prediksi. Metrik ini memberikan indikasi seberapa tidak pasti model dalam membuat prediksi.

```
def perplexity(y_true, y_pred):
   return np.exp(-np.mean([np.log(max(1e-10, p)) for p in y_pred]))

perplexity_score = perplexity(y_test, y_pred)
```

3.7 Perbandingan Model

Pada tahap ini, dilakukan perbandingan kinerja antara tiga model yang berbeda, yaitu model LSA (*Latent Semantic Analysis*), model *Transformer* , dan model *Hybrid*. Perbandingan ini bertujuan untuk mengevaluasi keunggulan dan kelemahan masing-masing model berdasarkan berbagai metrik evaluasi yang telah dihitung sebelumnya. Berikut adalah langkah-langkah yang dilakukan dalam tahap perbandingan:

a. Perbandingan Kinerja Model

Perbandingan ini bertujuan untuk mengevaluasi keunggulan dan kelemahan masing-masing model berdasarkan berbagai metrik evaluasi yang telah dihitung sebelumnya. Pada bagian ini, hasil evaluasi model *hybrid* dibandingkan dengan pendekatan individu, yaitu LSA dan *Transformer*. Tabel perbandingan dibuat untuk menyajikan kinerja ketiga model secara berdampingan.

Tabel 9. Perbandingan Kinerja Model

Model	***Accuracy***	***F1-Score***	***BLEU Score***	**ROUGE-1**	**ROUGE-L**	***Perplexity***
LSA	0.501626	0.464304	9.138776e-232	0.501626	0.501626	512698.689999

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Transformer	0.486143	0.495209	8.856715e-232	0.486143	0.486143	78613.052248
Hybrid	0.510760	0.520486	9.305192e-232	0.510760	0.510760	67440.516169

Pada Tabel 9 menunjukan bahwa model model *LSA* memiliki *accuracy* 0.5016 dan *F1-Score* 0.4643, menunjukkan kemampuannya menangkap pola semantik global, tetapi kehilangan informasi sintaksis. *Transformer* memiliki *F1-Score* lebih tinggi (0.4952) dengan *accuracy* 0.4861, berkat mekanisme *self-attention* yang menangkap hubungan antar kata dalam kalimat. Namun, model ini masih bergantung pada data pelatihan yang besar. Model *Hybrid* menunjukkan hasil terbaik dengan *accuracy* 0.5108 dan *F1-Score* 0.5205, karena menggabungkan keunggulan *LSA* dalam memahami semantik global dan *Transformer* dalam menangkap konteks lokal. *Perplexity* yang lebih rendah (67,440.52) dibandingkan *Transformer* (78,613.05) dan *LSA* (512,698.69) menunjukkan bahwa model ini lebih stabil dalam memprediksi teks. Nilai *BLEU Score* yang rendah di semua model disebabkan oleh karakteristik bahasa Indonesia yang fleksibel dan struktur evaluasi *BLEU* yang lebih cocok untuk *machine translation* daripada klasifikasi teks. Secara keseluruhan, hasil ini menegaskan bahwa pendekatan *Hybrid* lebih efektif dibandingkan metode individual dalam memahami teks berita berbahasa Indonesia.

b. Perbandingan Model Lain

Untuk memahami keunggulan dan keterbatasan pendekatan *hybrid*, dilakukan perbandingan dengan model lain seperti BERT (Bidirectional Encoder Representations from Transformers) dan GPT. Model BERT terkenal dengan kemampuannya dalam memahami konteks melalui representasi berbasis perhatian diri (self-attention), sedangkan GPT (Generative Pre-trained Transformer) lebih unggul dalam tugas pemodelan bahasa secara autoregresif. Dengan membandingkan performa *hybrid* terhadap model-model ini, dapat diperoleh wawasan lebih dalam mengenai efektivitas pendekatan *hybrid* dalam berbagai aspek evaluasi. Berikut adalah hasil perbandingan tambahan dengan model lain:

Tabel 10. Perbandingan hasil dengan model lain

Model	***Accuracy***	***F1-Score***	***BLEU Score***	**ROUGE-1**	**ROUGE-L**	***Perplexity***
BERT	0.543829	0.562198	15.217364	0.553104	0.551987	59823.427312
GPT	0.573891	0.582354	16.483276	0.572648	0.563219	57812.645891
Hybrid	0.510760	0.520486	9.305192e-232	0.510760	0.510760	67440.516169

Hasil pada Tabel 10 bahwa model BERT dan GPT memiliki kinerja yang lebih unggul dibandingkan model *hybrid*, terutama dalam *BLEU Score* dan *Perplexity*. BERT memiliki akurasi sebesar 0.543829 dan *F1-Score* 0.562198, sedangkan GPT mencapai akurasi 0.573891 dan *F1-Score* 0.582354, lebih tinggi dibandingkan dengan model *hybrid* yang memiliki akurasi 0.510760 dan *F1-Score* 0.520486. Dalam *BLEU Score*, BERT memperoleh 15.217364 dan GPT memperoleh 16.483276, jauh lebih tinggi dibandingkan model *hybrid* yang hanya mencapai 9.305192e-232. Selain itu, *Perplexity* dari model *hybrid* masih lebih tinggi (67440.516169) dibandingkan dengan BERT (59823.427312) dan GPT (57812.645891), menunjukkan bahwa model deep learning berbasis *Transformer* memiliki pemahaman yang lebih baik terhadap distribusi kata dalam teks. Namun, meskipun model *hybrid* memiliki performa lebih rendah dibandingkan BERT dan GPT, keunggulannya terletak pada efisiensi komputasi, menjadikannya pilihan yang lebih ringan untuk aplikasi dengan keterbatasan infrastruktur.

3.8 Keterbatasan Penelitian

Meskipun penelitian ini menunjukkan peningkatan dalam pemahaman teks berita berbahasa Indonesia melalui model *hybrid*, terdapat beberapa keterbatasan yang perlu diperhatikan:

a. Keterbatasan Dataset

Dataset yang digunakan berasal dari sumber berita tertentu, sehingga kemungkinan terdapat bias dalam struktur dan gaya bahasa yang digunakan. Selain itu, ukuran dataset masih terbatas, yang dapat mempengaruhi generalisasi model terhadap teks dengan domain atau gaya bahasa yang berbeda.

b. Keterbatasan Metode

Model *hybrid* yang diusulkan menggabungkan *Latent Semantic Analysis (LSA)* dan *Transformer* untuk menangkap hubungan semantik secara lebih baik. Namun, integrasi kedua pendekatan ini meningkatkan kompleksitas komputasi, terutama dalam proses *training* dan *inference*. Model ini membutuhkan sumber daya komputasi yang lebih besar dibandingkan dengan pendekatan individual, yang dapat menjadi kendala dalam implementasi skala besar.

c. Evaluasi Model

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Metrik evaluasi yang digunakan, seperti *Accuracy*, *F1-Score*, *BLEU Score*, dan *ROUGE Score*, memberikan gambaran performa model secara kuantitatif. Namun, metrik ini belum sepenuhnya mencerminkan pemahaman semantik model secara mendalam. Diperlukan analisis lebih lanjut, misalnya melalui *human evaluation*, untuk menilai kualitas hasil yang dihasilkan oleh model.

d. Generalizability

Model ini dikembangkan khusus untuk teks berita dalam bahasa Indonesia. Kemampuan model untuk diterapkan pada jenis teks lain, seperti opini, dokumen hukum, atau media sosial, belum dieksplorasi secara mendalam. Penelitian lebih lanjut diperlukan untuk menguji keandalan model dalam berbagai konteks bahasa.

e. Komparasi dengan Model Lain

Meskipun model *hybrid* menunjukkan peningkatan dalam pemahaman semantik, performanya masih perlu dibandingkan secara lebih mendalam dengan model yang lebih kompleks seperti *BERT* atau *GPT*. Model berbasis *deep learning* ini telah terbukti unggul dalam banyak tugas *Natural Language Processing (NLP)*, terutama dalam memahami konteks yang lebih kompleks. Diperlukan penelitian lebih lanjut untuk mengevaluasi bagaimana model *hybrid* dapat dikombinasikan atau dibandingkan dengan pendekatan yang lebih canggih ini guna meningkatkan kinerja secara keseluruhan.

4. KESIMPULAN

Model *hybrid* menunjukkan peningkatan dalam memahami konteks teks berita berbahasa Indonesia dengan menggabungkan kekuatan *LSA* dan *Transformer*. Pendekatan ini menghasilkan akurasi sebesar **0.510760** dan *F1-Score* sebesar **0.520486**, lebih tinggi dibandingkan model *LSA* dan *Transformer* secara individual. Namun, dibandingkan dengan *BERT* (**0.543829**, **0.562198**) dan *GPT* (**0.573891**, **0.582354**), model *hybrid* masih tertinggal dalam beberapa aspek, terutama pada *BLEU Score* yang hanya mencapai **9.305192e-232**, jauh di bawah *BERT* (**15.217364**) dan *GPT* (**16.483276**). Meski begitu, model *hybrid* memiliki keunggulan dalam efisiensi komputasi, menjadikannya pilihan yang lebih ringan untuk aplikasi dengan keterbatasan sumber daya.

Untuk meningkatkan kinerja model, penelitian selanjutnya dapat mempertimbangkan penggunaan dataset yang lebih besar guna meningkatkan kemampuan model dalam menangkap pola bahasa yang lebih luas. Selain itu, eksplorasi terhadap teknik *embedding* lain seperti *word2vec*, *fastText*, atau *contextual embeddings* dapat membantu meningkatkan representasi kata. Evaluasi pada berbagai domain teks juga diperlukan untuk melihat sejauh mana model dapat digeneralisasi, sementara perbandingan dan integrasi dengan model *deep learning* yang lebih kompleks seperti *BERT*, *GPT*, atau *T5* dapat menjadi langkah selanjutnya dalam menyempurnakan pendekatan *hybrid* ini.

DAFTAR PUSTAKA

[1] C. Primasiwi, M. I. Irawan, and R. Ambarwati, “Key Performance Indicators for Influencer Marketing on Instagram:,” presented at the 2nd International Conference on Business and Management of Technology (ICONBMT 2020), Surabaya, Indonesia, 2021. doi: 10.2991/aebmr.k.210510.027.

[2] T. Guarda, J. Balseca, K. García, J. González, F. Yagual, and H. Castillo-Beltran, “Digital Transformation Trends and Innovation,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1099, no. 1, p. 012062, Mar. 2021, doi: 10.1088/1757-899X/1099/1/012062.

[3] A. Dewandaru, D. H. Widyantoro, and S. Akbar, “Event Geoparser with Pseudo-Location Entity Identification and Numerical Extraction in Indonesian News Corpus,” Aug. 14, 2020, *MATHEMATICS & COMPUTER SCIENCE*. doi: 10.20944/preprints202008.0263.v1.

[4] A. F. Aji *et al.*, “One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 7226–7249. doi: 10.18653/v1/2022.acl-long.500.

[5] M. Hahn, “Theoretical Limitations of Self-Attention in Neural Sequence Models,” *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 156–171, Dec. 2020, doi: 10.1162/tacl_a_00306.

[6] T. Wolf *et al.*, “Transformers: State-of-the-Art Natural Language Processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, 2020, pp. 38–45. doi: 10.18653/v1/2020.emnlp-demos.6.

[7] D. F. O. Onah, E. L. L. Pang, and M. El-Haj, “A Data-driven Latent Semantic Analysis for Automatic Text Summarization using LDA Topic Modelling,” in *2022 IEEE International Conference on Big Data (Big Data)*, Osaka, Japan: IEEE, Dec. 2022, pp. 2771–2780. doi: 10.1109/BigData55660.2022.10020259.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

[8] D. Patterson *et al.*, "Carbon Emissions and Large Neural Network Training," 2021, *arXiv*. doi: 10.48550/ARXIV.2104.10350.

[9] Y. Li and A. Risteski, "The Limitations of Limited Context for Constituency Parsing," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, 2021, pp. 2675–2687. doi: 10.18653/v1/2021.acl-long.208.

[10] C. W. Schmidt *et al.*, "Tokenization Is More Than Compression," Oct. 07, 2024, *arXiv*: arXiv:2402.18376. doi: 10.48550/arXiv.2402.18376.

[11] M. Amien, "Sejarah dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia: Tinjauan tentang sejarah, perkembangan teknologi, dan aplikasi NLP dalam bahasa Indonesia," Mar. 28, 2023, *arXiv*: arXiv:2304.02746. doi: 10.48550/arXiv.2304.02746.

[12] R. Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the Accuracy of Text Classification using Stemming Method, A Case of Non-formal Indonesian Conversation".

[13] H. T. Y. Achsan, H. Suhartanto, W. C. Wibowo, D. A. Dewi, and K. Ismed, "Automatic Extraction of Indonesian Stopwords," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 2, 2023, doi: 10.14569/IJACSA.2023.0140221.

[14] G. Garrido-Bañuelos, Mpho Mafata, and A. Buica, "Exploring the use of Latent Semantic Analysis (LSA) to investigate wine sensory profiles," 2024, doi: 10.13140/RG.2.2.32030.96328.

[15] T. Xiao and J. Zhu, "Introduction to Transformers: an NLP Perspective," Nov. 29, 2023, *arXiv*: arXiv:2311.17633. doi: 10.48550/arXiv.2311.17633.

[16] T. Q. Nguyen, K. Murray, and D. Chiang, "Data Augmentation by Concatenation for Low-Resource Translation: A Mystery and a Solution," Jul. 02, 2021, *arXiv*: arXiv:2105.01691. doi: 10.48550/arXiv.2105.01691.

[17] A. Galassi, M. Lippi, and P. Torroni, "Attention in Natural Language Processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021, doi: 10.1109/TNNLS.2020.3019893.

[18] M. Pagliardini, A. Mohtashami, F. Fleuret, and M. Jaggi, "DenseFormer: Enhancing Information Flow in Transformers via Depth Weighted Averaging," Mar. 21, 2024, *arXiv*: arXiv:2402.02622. doi: 10.48550/arXiv.2402.02622.

[19] K. M. Kahloot and P. Ekler, "Algorithmic Splitting: A Method for Dataset Preparation," *IEEE Access*, vol. 9, pp. 125229–125237, 2021, doi: 10.1109/ACCESS.2021.3110745.

[20] Ž. Đ. Vujovic, "Classification Model Evaluation Metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120670.

[21] "CLINICAL DETERIORATION PREDICTION IN BRAZILIAN HOSPITALS BASED ON ARTIFICIAL NEURAL NETWORKS AND TREE DECISION MODELS," in *Proceedings of the 15th International Conference on ICT, Society and Human Beings (ICT 2022), the 19th International Conference Web Based Communities and Social Media (WBCSM 2022) and 14th International Conference on e-Health (EH 2022)*, IADIS Press, Jul. 2022. doi: 10.33965/ICT_WBC_EH2022_202204L024.

[22] J. Wieting, T. Berg-Kirkpatrick, K. Gimpel, and G. Neubig, "Beyond BLEU: Training Neural Machine Translation with Semantic Similarity," Sep. 14, 2019, *arXiv*: arXiv:1909.06694. doi: 10.48550/arXiv.1909.06694.

[23] A. Bharadwaj, A. Srinivasan, A. Kasi, and B. Das, "Extending The Performance of Extractive Text Summarization By Ensemble Techniques," in *2019 11th International Conference on Advanced Computing (ICoAC)*, Chennai, India: IEEE, Dec. 2019, pp. 282–288. doi: 10.1109/ICoAC48765.2019.246854.

[24] J. Roh, S.-H. Oh, and S.-Y. Lee, "Unigram-Normalized Perplexity as a Language Model Performance Measure with Different Vocabulary Sizes," Nov. 26, 2020, *arXiv*: arXiv:2011.13220. doi: 10.48550/arXiv.2011.13220.