p-ISSN: 2301-5373
e-ISSN: 2654-5101

Pengaruh Augmentasi Teks pada Klasifikasi Berita Menggunakan BERT *Embedding* dan LSTM

Ni Made Gita Satviki Nirmala[a1], Ngurah Agus Sanjaya ER[a2],
I Dewa Made Bayu Atmaja Darmawan[a3], I Made Widiartha[a4]

[a]Program Studi Informatika, Universitas Udayana
Bali, Indonesia
[1]nirmala.2208561053@unud.ac.id
[2]agus_sanjaya@unud.ac.id
[3]dewabayu@unud.ac.id
[4]madewidiartha@unud.ac.id

Abstract

According to the 2023 Digital News Report, 84% of Indonesians access or consume news online, reflecting how central digital platforms have become to public information consumption. Furthermore, 2024 data from the Indonesian Press Council recorded 1,015 online news platforms operating in Indonesia. With such a diverse range of providers, it is estimated that thousands or even millions of news articles are published annually, making consistent and efficient content organization increasingly important. News category labels serve as a universal variable that helps both online news platform providers and the general public navigate this volume of information more easily. This study therefore performs automatic news topic classification by combining BERT embedding with Long Short-Term Memory (LSTM), incorporating back-translation as a data augmentation technique to mitigate overfitting. Grid search hyperparameter tuning was employed to determine the optimal configuration, resulting in a dropout rate of 0.2 and a learning rate of 0.0001 for both the non-augmented and 30%-augmented models. The non-augmented model achieved 96% accuracy, while the 30%-augmented model achieved 98% accuracy, establishing it as the best-performing approach in this study.

Keywords : *Back translation, BERT Embedding, Grid search, Long Short Term Memory, News Classification*

1. Pendahuluan

Berita dapat diartikan sebagai cerita atau keterangan mengenai kejadian atau peristiwa yang hangat [1]. Menurut digital *news report* pada tahun 2023 diketahui bahwa sebanyak 84% masyarakat Indonesia membaca atau mengetahui berita secara *online* atau daring [2]. Dari data tersebut dapat diketahui bahwa masyarakat Indonesia lebih menikmati berita melalui secara *online* atau daring. Fakta ini didukung oleh data dari dewan pers Indonesia pada tahun 2024 terdapat sebanyak 1.015 *platform* penyedia berita *online* [3]. Dengan beragamnya *platform* penyedia berita dapat diperkirakan terdapat ribuan atau mungkin jutaan berita yang ada di setiap tahunnya. Untuk memudahkan pembaca dalam mengakses berita diperlukan variabel berupa label kategori berita yang diaplikasikan untuk seluruh berita. Oleh karena itu, dibutuhkan sebuah sistem yang mampu melakukan klasifikasi topik pada berita untuk membantu dan memudahkan penyedia *platform* berita maupun masyarakat awam.

Beberapa penelitian terdahulu telah melakukan penelitian mengenai klasifikasi teks dengan berbagai pendekatan. Salah satunya penelitian yang membahas mengenai masalah yang sering terjadi pada NLP berupa data latih yang terbatas dan terjadinya *overfitting*, sehingga dilakukan pendekatan SSL-Reg berbasis *self-supervised learning*. Dari hasil penelitian tersebut

didapatkan hasil bahwa pendekatan yang dilakukan dapat menangani perihal *overfitting* yang terjadi, serta untuk pendekatan tersebut merupakan sama dengan BERT [4]. Selanjutnya terdapat penelitian mengenai klasifikasi teks yang dilakukan dengan menggunakan model LSTM dan BiLSTM untuk melakukan pengujian dari *dataset* yang kecil. Hasil yang didapatkan untuk f1-score BiLSTM yaitu 94,15% sedangkan untuk LSTM di 93,16%, tetapi dikarenakan *dataset* yang minim maka terjadi *overfitting* ringan [5]. Serta terdapat penelitian yang membahas mengenai klasifikasi teks yang difokuskan untuk pendeteksian judul *clickbait* dan komentar *hate speech* pada berita *online* dengan menggunakan SVM, IndoBERT dan augmentasi data. Pada pengujian menghasilkan data bahwa augmentasi data mampu meningkatkan skor F1 sekitar 0,1% hingga 8% [6].

Beragamnya penelitian yang telah dilakukan peneliti terdahulu menunjukkan hasil yang baik, namun terdapat beberapa kekurangan atau peningkatan yang dapat dilakukan. Salah satu permasalahan yang sering terjadi yaitu mengenai kurangnya *dataset* serta terjadinya kasus *overfitting*. Oleh karena itu, pada penelitian ini menggunakan augmentasi teks berupa *back translation* yang diharapkan dapat membantu dalam mencegah *overfitting* dan memperkaya *dataset*. Terdapat juga kombinasi model dengan menggunakan BERT *embedding* dan *Long Short Term Memory*. BERT *embedding* merupakan mekanisme yang dapat memahami konteks antar teks [7], sehingga diharapkan model menjadi lebih baik melakukan klasifikasi. Sedangkan *Long Short Term Memory* atau LSTM merupakan metode yang mampu mengingat informasi dalam jangka waktu panjang sekaligus membuang informasi yang tidak lagi relevan [8]. Serta untuk memaksimalkan model yang digunakan terdapat proses *hyperparamater tuning* dengan menggunakan *grid search*. Sehingga kombinasi akhir dari penelitian ini berupa penggunaan BERT *embedding* dan LSTM dengan terdapat augmentasi, serta proses *hyperparameter tuning* dengan *grid search*. Kombinasi tersebut dilakukan e*val*uasi dengan melihat nilai dari akurasi, *recall*, *precision*, dan *f1-score*.

2. Metode Penelitian

Penelitian ini dilakukan beberapa tahapan yaitu pengumpulan data, *exploratory data analysis*, verifikasi label dengan ahli pakar, *preprocessing*, augmentasi data dengan *back translation*, BERT *embedding*, *Long short Term Memory* untuk model utama, pencarian parameter terbaik dengan *hyperparameter tuning*, dan terakhir melakukan e*val*uasi model secara keseluruhan.

2.1 Pengumpulan Data

Dataset yang digunakan pada penelitian ini diperoleh melalui tahapan web *scraping*. *Scraping* dilakukan pada *platform* berita *online* terbesar di Indonesia seperti Kompas, CNN. dan Detik. *Scraping* dilakukan dengan mencari *dataset* rentang tahun 2024 serta label kategori “politik”, “ekonomi”, “kesehatan”, “olahraga”, dan “teknologi”. *Scraping* dilakukan dengan memanfaatkan *url website platform* serta *tag* dan *class* HTML yang didapatkan dengan cara *inspect website*. Terdapat pengaturan waktu serta penggunaan *user-agent* untuk mencegah sistem dideteksi sebagai bot atau robot. Atribut yang diperoleh melalui *scraping* berupa tanggal berita, judul berita, *link* berita, isi berita, sumber berita, dan kategori berita. Jumlah data yang diperoleh melalui *scraping* sebesar 20.611 data, dengan terdapat 4.778 label ekonomi, 4.038 label kesehatan, 5.312 label olahraga, 2.949 label politik, dan 3.534 label teknologi. Contoh data yang diperoleh dapat dilihat pada Tabel 1.

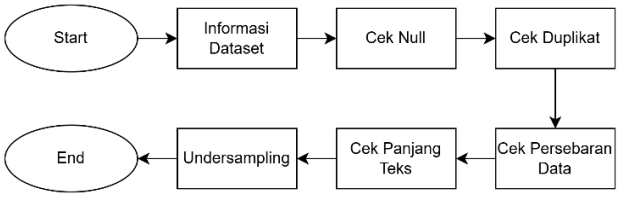
Tabel 1. Data Berita Hasil *Scraping*

tanggal	2024-01-01
judul	Marc Marquez, Pindah Tim dan Mengais Asa Kembali Jadi Juara MotoGP
link	https://www.cnnindonesia.com/olahraga/20231230203620-156-1043610/marc-marquez-pindah-tim-dan-mengais-asa-kembali-jadi-juara-motogp
isi	Marc Marquezresmi bergabung dengan Gresini Racing diMotoGP 2024. Bagaimana prediksi kiprah Marc Marquez di MotoGP 2024? Selain tim baru, pembalap asal Spanyol itu juga akan mengendarai sepeda motor anyar. Marc Marquez akan menunggangi Ducati untuk pertama kalinya dalam karier MotoGP usai satu dekade bersama Honda.....
sumber	CNN
kategori	olahraga

p-ISSN: 2301-5373
e-ISSN: 2654-5101

2.2 *Exploratory Data Analysis* (EDA)

Exploratory data analysis atau EDA merupakan suatu tahapan yang digunakan untuk membantu untuk melakukan analisis terhadap *dataset* dengan tujuan untuk menyelidiki dan memahami *dataset*. Pada penelitian ini dilakukan beberapa tahapan, pertama dilakukan pengecekan informasi *dataset* yang berfungsi untuk mengetahui jenis kolom yang terdapat dalam *dataset* serta jumlah dari *dataset* yang dimiliki. Tahap kedua yaitu melakukan cek *null* untuk mencegah terdapat data kosong atau null yang menyebabkan *noise* pada data. Tahap ketiga dilakukan pengecekan duplikat, tahap ini dilakukan untuk membantu dalam melihat apakah terdapat data ganda pada *dataset* yang dapat mempengaruhi performa model. Tahap keempat dilakukan pengecekan persebaran data, tujuan dari proses ini untuk memastikan *dataset* sudah memiliki label yang sesuai. Tahap kelima yang dilakukan yaitu pengecekan panjang teks yang dilakukan untuk memastikan data hasil *scraping* memiliki kualitas yang baik. Serta tahap terakhir yang dilakukan berupa melakukan *undersampling* untuk membuat data seimbang. Untuk tahapan EDA dapat dilihat pada Gambar 1.



Gambar 1. Tahapan EDA

2.3 Verifikasi Label dengan Ahli Pakar

Dikarenakan *dataset* pada penelitian ini merupakan *dataset* primer yang diperoleh melalui proses *scraping*, label yang didapatkan belum tentu benar dan diperlukan verifikasi ahli. Verifikasi yang digunakan pada penelitian ini melalui dua tahap yaitu verifikasi LLM dan verifikasi oleh ahli pakar. Proses pertama yang dilakukan merupakan verifikasi LLM dengan menggunakan API Grok. Verifikasi ini dipergunakan untuk membandingkan label yang diperoleh melalui *scraping* dan label yang didapati LLM, bila terdapat perbedaan maka akan terdapat keterangan FALSE. Data dengan perbedaan label dari hasil prediksi LLM maka melewati tahapan kedua dengan melakukan verifikasi dengan ahli pakar. Ahli pakar yang melakukan verifikasi dilakukan oleh Bapak Ida Bagus Gede Dharma Putra, S.S., M.A. yang merupakan Dosen Fakultas Ilmu Budaya, Program Studi Sastra Indonesia, Universitas Udayana. Setelah melalui tahapan verifikasi dengan LLM dan ahli pakar, menyebabkan *dataset* menjadi tidak seimbang kembali. Oleh karena itu, dilakukan kembali tahapan *undersampling* dengan 1.000 data untuk setiap kategori. Sehingga *dataset* akhir yang digunakan sebanyak 5.000 data.

2.4 *Preprocessing*

Preprocessing merupakan suatu tahapan yang dilakukan untuk mempersiapkan data sebelum masuk ke dalam model [9]. Persiapan data ini dipergunakan untuk memaksimalkan hasil dari model. Pada penelitian ini dilakukan *preprocessing* data hanya berupa *cleaning* data. Hal ini dikarenakan pada penelitian menggunakan BERT *Embedding* yang membutuhkan konteks kata atau semantik untuk mengerti data. Oleh sebab itu, hanya dilakukan tahapan *cleaning* data yang membantu *dataset* bersih sehingga menghasilkan model yang optimal. Pada penelitian ini proses *cleaning* data meliputi mengubah teks ke *lowercase*, mengubah tanda \r\n atau *newline* menjadi spasi, menghapus *url*, Menghapus *tag html*, Menghapus karakter spesial, menghapus spasi berlebih, menghapus teks *scroll to continue with content*, menghapus teks yang terdapat teks Gambas:, menghapus teks Jakarta, kompas.com, menghapus teks kompas.com serta menghapus teks yang terdapat teks *freepik, unsplash, shutterstock, pexels, istockphoto, gettyimage*, kompas.id. *Cleaning* tersebut juga diatur agar tidak memperhatikan kapitalisasi pada data, sehingga *lowercase* dan *uppercase* memiliki makna yang sama.

2.5 Augmentasi

Augmentasi merupakan suatu proses yang bertujuan untuk mengurangi terjadinya *overfitting* pada saat pemodelan, serta untuk mengatasi masalah kurangnya data dengan menghasilkan lebih banyak data berdasarkan korpus [10]. Augmentasi yang digunakan pada penelitian ini berupa *back translation*. *Back translation* merupakan salah satu teknik augmentasi dengan menerjemahkan data ke bahasa tertentu dan dikembalikan kembali ke bahasa asli [11]. Dalam melakukan tahapan augmentasi dilakukan beberapa rasio untuk membandingkan rasio terbaik. Rasio yang digunakan berupa 10%, 20%, 30%, 40%, dan 50%. Augmentasi dilakukan dengan mengambil data dari setiap kategori secara acak dan sesuai dengan rasio yang ditetapkan.

2.6 BERT *Embedding*

Setelah melalui tahapan augmentasi atau tanpa melalui augmentasi, *dataset* dibagi menjadi 70% *training* dan 30% *testing* yang selanjutnya data masuk ke dalam tahapan BERT *embedding*. BERT *embedding* merupakan salah satu penerapan *word embedding* yang merupakan metode untuk membuat vektor dengan mengubah representasi kata, memungkinkan kata-kata dengan makna yang sama memiliki representasi yang mirip [12]. Pada saat memasuki tahapan BERT *embedding*, hal yang dilakukan pertama kali yaitu tahap *wordpiece tokenizer*. Proses *wordpiece tokenizer* dibagi menjadi tiga tahapan yaitu mengubah data menjadi bentuk token, membaca kamus yang tersedia, dan memisahkan kata dengan acuan kamus dan menambahkan "##" bila terdapat kata yang tidak ada pada kamus. Untuk kamus yang digunakan berupa IndoBERT yaitu "indobenchmark/indobert-base-p1". Setelah proses *wordpiece tokenizer* maka selanjutnya menambahkan token [CLS] di setiap awal data dan token [SEP] untuk setiap akhir data. Baru dilakukan konversi token id dan masuk ke tahapan *embedding* utama. Pada tahapan *embedding* dibagi menjadi token *embedding*, segment *embedding*, dan position *embedding*.

2.7 Long Short Term Memory

Hasil dari *embedding* selanjutnya diproses ke dalam arsitektur LSTM untuk menghasilkan klasifikasi berita. LSTM adalah metode yang mampu mengingat informasi dalam jangka waktu panjang sekaligus membuang informasi yang tidak lagi relevan [8]. Pada arsitektur LSTM terdapat beberapa komputasi dengan memanfaatkan *hidden state*, *cell state*, *forget gate*, *input gate*, *cell candidate*, dan *output gate*. *Hidden state* h(t-1) berguna untuk menyimpan informasi dari langkah waktu sebelumnya yang akan digunakan dengan input. *Cell state* c(t-1) berfungsi untuk menyimpan memori jangka panjang dari langkah waktu sebelumnya. *Forget gate* berfungsi untuk menentukan informasi mana dari *cell state* sebelumnya yang perlu dilupakan atau yang perlu dipertahankan. *Input gate* berfungsi untuk menentukan informasi baru mana yang diperlukan untuk disimpan ke dalam *cell state*. *Cell candidate* merupakan menghasilkan kandidat nilai baru yang akan ditambahkan ke *cell state*. *Update cell state* diperlukan untuk memperbarui *cell state* dengan menggabungkan informasi yang dipertahankan dari *forget gate* dan informasi baru dari *input gate*. *Output gate* memiliki fungsi untuk menentukan bagian mana dari *cell state* yang akan menjadi *output*. Serta yang terakhir merupakan *hidden state* yang menghasilkan *hidden state* baru sebagai *output* akhir untuk diteruskan ke langkah berikutnya. Setelah itu masuk ke tahapan *dropout layer* dan *dense layer*, sehingga mendapatkan hasil prediksi dari klasifikasi berita. Untuk rumus yang digunakan dalam tahapan LSTM sebagai berikut:

$$forget\ gate\ (f_t) = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \quad (1)$$

$$input\ gate\ (i_t) = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$cell\ candidate\ (C_t) = tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$

$$update\ cell\ state\ (C_t) = f_t * C_{t-1} + i_t * C_t \quad (4)$$

$$output\ gate\ (o_t) = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$hidden\ state\ (h_t) = o_t * tanh(C_t) \quad (6)$$

p-ISSN: 2301-5373
e-ISSN: 2654-5101

2.8 *Hyperparameter tuning*

Hyperparameter tuning merupakan merupakan suatu parameter yang digunakan sebelum proses pembelajaran model dilakukan pada machine learning atau deep learning [13]. Jens *hyperparameter tuning* yang digunakan yaitu dengan metode *grid search*, dikarenakan metode ini melakukan seluruh kombinasi yang ada sehingga mendapatkan hasil yang paling terbaik. Pada penelitian ini pencarian *hyperparameter tuning* difokuskan pada parameter *dropout* dan *learning rate*. Untuk penelitian ini terdapat 12 kombinasi yang dilakukan, dengan nilai uji yang terdapat pada Tabel 2.

Tabel 2. Parameter *Hyperparameter tuning*

Parameter	Nilai Uji
dropout	0,2; 0,3; 0,4; 0,5
learning rate	0,0001; 0,001; 0,0005

2.9 E*val*uasi Model

E*val*uasi model adalah proses untuk memastikan kualitas dan keandalan, serta mengukur kinerja dan kemampuan model yang dikembangkan dalam membuat prediksi atau menghasilkan *output* yang akurat berdasarkan data yang telah dipelajari [14]. Pada penelitian ini dilakukan e*val*uasi berupa melihat nilai akurasi, *precision, recall,* dan *f1-score*. Berikut rumus yang digunakan untuk setiap metrik e*val*uasi beserta keterangan simbolnya *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$precision = \frac{TP}{TP+FP} \quad (8)$$

$$f1-score = 2 \times \frac{precision \times recall}{precision+recall} \quad (9)$$

$$recall = \frac{TP}{TP+FN} \quad (10)$$

3. Hasil dan Pembahasan

3.1 Hasil Augmentasi

Augmentasi dilakukan dengan melakukan *back translation* yang memanfaatkan API *google translate* pada *library deep_translator*. Augmentasi dilakukan dengan mengubah bahasa utama *dataset* yaitu bahasa Indonesia menjadi bahasa Inggris, yang selanjutnya dikembalikan ke bahasa asli bahasa Indonesia. Augmentasi dilakukan dengan membagi data ke masing-masing kategori, baru dilakukan augmentasi sesuai besar augmentasi yang dipilih. Untuk detail data setelah dilakukan augmentasi dapat dilihat pada Tabel 3.

Tabel 3. Jumlah Data Augmentasi

rasio	jumlah per kategori	jumlah keseluruhan
10%	100	500
20%	200	1000
30%	300	1500
40%	400	2000
50%	500	2500

Pada penerapan dalam kode terdapat pengaturan waktu untuk menunggu dari *translate* ke bahasa Inggris dan mengembalikan kembali ke bahasa Indonesia untuk mencegah terjadinya limit. Serta terdapat pengecekan agar hasil augmentasi tidak kosong atau *null*, hasil memiliki jumlah karakter lebih dari 10, dan hasil augmentasi tidak sama dengan teks asli. Bila hasil augmentasi memenuhi syarat tersebut baru akan disimpan dan melakukan proses augmentasi selanjutnya.

3.2 Hasil BERT *Embedding*

Sebelum masuk ke tahapan BERT *embedding* data hasil augmentasi atau tanpa augmentasi dilkukan split data, dengan perbandingan *training* 70%, test 15%, dan *val* 15%. Selanjutnya data memasuki tahapan *embedding* dengan melewati beberapa tahapan yaitu *wordpiece tokenizer*, penambahan token [CLS] dan [SEP], dan tahapan *embedding* utama. Pada tahapan *embedding* utama dibagi menjadi tiga tahapan yaitu token *embedding*, segment *embedding*, dan position *embedding* yang kemudian dijumlahkan untuk menghasilkan hasil akhir *embedding* berukuran 768 dimensi. Pada penelitian ini menggunakan padding sebesar 64. Serta terdapat pengaturan untuk membekukan BERT, hal ini diperlukan untuk meringankan komputasi. Sehingga BERT digunakan sebagai feature extractor yang menghasilkan *embedding* saat terdapat input baru, tetapi bobot token lainnya tidak diubah. Adapun contoh penerapan *wordpiece tokenizer* pada data seperti Tabel 4.

Tabel 4. Penerapan *Wordpiece tokenizer*

[CLS] wakil menteri badan usaha milik negara bumn kartika wir ##joat ##modjo memastikan pihak ##nya akan melakukan pendekatan hukum terkait dengan kasus yang terjadi di pt indo ##farma tbk [SEP]

3.3 Hasil Model LSTM dan *Hyperparameter tuning*

Hyperparameter difokuskan untuk mencari nilai *dropout* (0,2; 0,3; 0,4; 0,5) dan *learning rate* (0,0001; 0,001; 0,0005) yang menghasilkan 12 kombinasi. Untuk 5 kombinasi terbaik dilakukan kembali proses *k-fold cross validation* dengan 3 fold, tahapan ini dilakukan untuk menentukan kombinasi yang memiliki performa terbaik. Adapun tiga kombinasi terbaik setelah *k-fold cross validation* untuk model tanpa augmentasi dan augmentasi seperti pada Tabel 5.

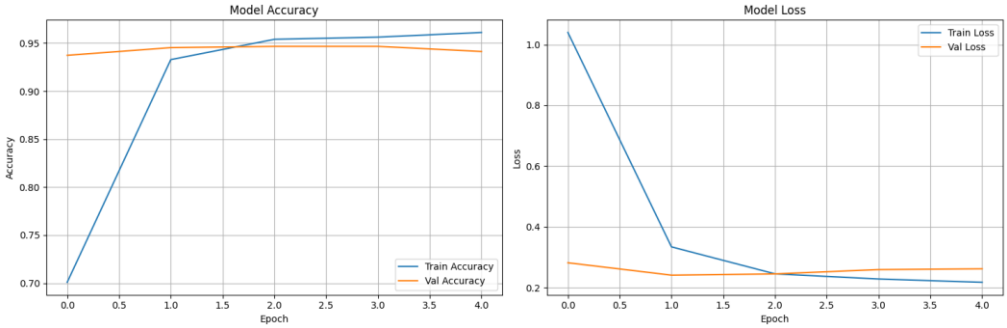
Tabel 5. Tiga Kombinasi Terbaik Model Augmentasi dan Tanpa Augmentasi

model	*dropout*	*learning rate*	akurasi
Tanpa Augmentasi	0,2	0,0001	0,961
	0,3	0,001	0,960
	0,4	0,0005	0,960
Augmentasi 10%	0,2	0,0001	0,961
	0,3	0,001	0,960
	0,4	0,0005	0,960
Augmentasi 20%	0,2	0,0001	0,961
	0,2	0,0005	0,960
	0,3	0,0001	0,958
Augmentasi 30%	0,2	0,0001	0,965
	0,2	0,001	0,965
	0,2	0,0005	0,965
Augmentasi 40%	0,3	0,0005	0,966
	0,3	0,001	0,966
	0,2	0,0005	0,964
Augmentasi 50%	0,2	0,001	0,972
	0,2	0,0001	0,969
	0,3	0,0005	0,968

p-ISSN: 2301-5373
e-ISSN: 2654-5101

3.4 Hasil E*val*uasi Model

Model tanpa augmentasi menggunakan parameter *dropout* 0,2 dan *learning rate* 0,0001 sesuai dengan hasil pengujian. Hasil yang didapatkan oleh model tanpa augmentasi mendapatkan nilai akurasi untuk *train* yaitu 96%, sedangkan nilai akurasi untuk *val* yaitu 94,2%. Terdapat perbedaan atau *gap* sebesar 2,1%. Sedangkan untuk nilai *loss* pada *train* sebesar 0,21 dan pada *val* yaitu 0,26. Terdapat *gap* atau jarak sebesar 0,05 untuk *loss*. Jumlah *epoch* yang ada pada model yaitu 5 *epoch*. Dari informasi tersebut diketahui terdapat sedikit *overfitting*, dikarenakan terdapat pola *train loss* menurun sedangkan *val loss* stagnan. Akurasi akhir yang didapatkan model tanpa augmentasi dari hasil *testing* yaitu sebesar 96%. Untuk diagram perbandingan akurasi dan *loss* untuk test dan *val* dapat dilihat pada Gambar 2.



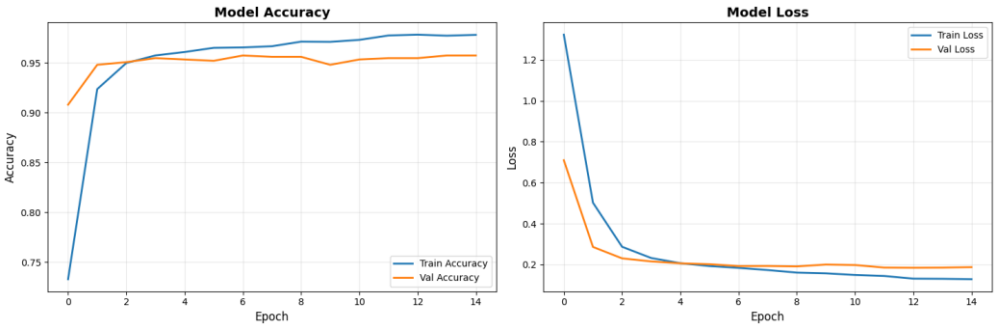
Gambar 2. Diagram Perbandingan *Train* dan *Val* Tanpa Augmentasi

Sedangkan untuk model dengan augmentasi mendapatkan hasil seperti pada Tabel 6. Untuk model dengan augmentasi 10% terdapat *gap* atau perbedaan sebesar 3% untuk nilai akurasi dan *gap* 0,105 untuk nilai *loss*. Sedangkan untuk model dengan augmentasi 20% terdapat *gap* sebesar 3,5% untuk nilai akurasi dan *gap* 0,10 untuk nilai *loss*. Model dengan augmentasi 30% memiliki *gap* atau perbedaan sebesar 2% untuk nilai akurasi dan *gap* 0,055 untuk nilai *loss*. Untuk model dengan augmentasi 40% memiliki perbedaan atau *gap* sebesar 2,5% untuk nilai akurasi dan *gap* 0,10 untuk nilai *loss*. Serta model dengan augmentasi 50% memiliki *gap* 2% untuk nilai akurasi dan 0,085 untuk nilai *loss*.

Tabel 6. Perbandingan Nilai Model dengan Augmentasi

Jumlah Augmentasi	Nilai Akurasi		Nilai *Loss*	
	Train	***Val***	***Train***	***Val***
10%	98,5%	95,5%	0,010	0,205
20%	98,7%	95,2%	0,11	0,21
30%	97,8%	95,8%	0,13	0,185
40%	99,5%	97%	0,045	0,145
50%	98,4%	96,4%	0,145	0,145

Dari seluruh perbandingan pada tabel di atas, diketahui bahwa model paling baik untuk model dengan augmentasi yaitu model augmentasi 30%. Hal ini dikarenakan nilai *val loss* yang sudah pada titik terendah dan stabil. Sehingga diketahui bahwa variasi data cukup untuk menambah data pada model. Serta nilai *gap loss* pada model ini termasuk kecil yang menandakan terdapat *overfitting* ringan tetapi masih dalam batas wajar dan memiliki performa paling baik dari seluruh model baik yang augmentasi maupun tidak augmentasi. Parameter yang digunakan untuk augmentasi 30% yaitu *dropout* di 0,2 dan *learning rate* di 0,0001. Jumlah *epoch* yang dilakukan sebanyak 14 *epoch*. Untuk nilai akurasi *testing* yang diperoleh model augmentasi 30% yaitu sebesar 98%. Adapun diagram perbandingan *train* dan *val* untuk augmentasi 30% dapat dilihat pada Gambar 3.



Gambar 3. Diagram Perbandingan *Train* dan *Val* Augmentasi 30%

4. Kesimpulan

Dari hasil penelitian yang dilakukan didapatkan kesimpulan untuk hasil akurasi yang didapatkan dari klasifikasi menggunakan BERT *embedding* dan LSTM tanpa menggunakan augmentasi yaitu sebesar 96%. Sedangkan hasil akurasi yang didapatkan dari model yaitu untuk augmentasi 10% mendapatkan akurasi 98%, augmentasi 20% mendapatkan hasil 98%, augmentasi 30% mendapatkan hasil 98%, augmentasi 40% mendapatkan hasil 99%, dan augmentasi 50% mendapatkan hasil 97%. Dengan model augmentasi yang paling baik yaitu augmentasi 30% *dataset*. Sedangkan untuk kombinasi terbaik yang didapatkan dari *hyperparameter tuning* untuk model tanpa augmentasi dan augmentasi terbaik di 30% sama-sama mendapatkan nilai *dropout* 0,2 dan *learning rate* sebesar 0,0001.

Hasil tersebut menunjukkan bahwa penelitian yang dilakukan berhasil untuk mengurangi *overfitting* yang terjadi pada model. Walaupun mendapatkan hasil yang tidak terlalu jauh, tetapi *back translation* pada penelitian ini membantu model untuk memiliki performa maksimal di augmentasi 30%. Saran untuk peneliti selanjutnya dapat melakukan penelitian dengan klasifikasi multi *labeling* untuk mendapatkan performa klasifikasi yang lebih baik lagi.

References

[1] "Berita," KBBI Daring. [Online]. Available: https://kbbi.kemdikbud.go.id/entri/berita

[2] J. Steele, "Digital News Report Indonesia 2023," Jun. 14, 2023. [Online]. Available: https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2023/indonesia

[3] A. Sapto Anggoro, "SURVEI LANSKAP MEDIA PERS INDONESIA," *Dewan Pers*, Jun. 12, 2024.

[4] M. Zhou, Z. Li, and P. Xie, "Self-supervised regularization for text classification," *Trans. Assoc. Comput. Linguist.*, vol. 9, pp. 641–656, 2021.

[5] C. Liu, "Long short-term memory (LSTM)-based news classification model," *PLoS One*, vol. 19, no. 5, p. e0301835, 2024.

[6] I. A. Rahma and L. H. Suadaa, "Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 6, pp. 1329–1340, 2023.

[7] T. I. Z. M. Putra, S. Suprapto, and A. F. Bukhori, "Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia," *Jurnal Ilmu Siber dan Teknologi Digital*, vol. 1, no. 1, pp. 1–28, 2022.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

[8] B. A. H. Kholifatullah and A. Prihanto, “Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech,” *Journal of Informatics and Computer Science (JINACS)*, pp. 292–297, 2023.

[9] R. L. Danardya and E. Winarko, “DATA AUGMENTATION EFFECT ON MULTI-LABEL EMOTION CLASSIFICATION OF INDONESIAN TWEETS USING SVM AND LSTM,” Universitas Gadjah Mada, 2022.

[10] K. Y. Shyang, “TEXT AUGMENTATION FOR EMOTION CLASSIFICATION IN MICROBLOG TEXT USING SIMILARITY SCORING BASED ON NEURAL EMBEDDING MODELS,” Universitas Sains Malaysia, 2022.

[11] K. Y. Shyang, “TEXT AUGMENTATION FOR EMOTION CLASSIFICATION IN MICROBLOG TEXT USING SIMILARITY SCORING BASED ON NEURAL EMBEDDING MODELS,” Universitas Sains Malaysia, 2022.

[12] S. Imron, E. I. Setiawan, and J. Santoso, “Deteksi Aspek Review E-Commerce Menggunakan IndoBERT Embedding dan CNN,” *INSYST: Journal of Intelligent System and Computation*, vol. 5, no. 1, pp. 10–16, 2023.

[13] M. Feurer and F. Hutter, “Hyperparameter optimization,” in *Automated machine learning: Methods, systems, challenges*, Springer, 2019, pp. 3–33.

[14] R. Merdiansah, S. Siska, and A. A. Ridha, “Analisis sentimen pengguna X Indonesia terkait kendaraan listrik menggunakan IndoBERT,” *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 221–228, 2024.