

Klasifikasi Cyberbullying dengan Menggunakan Convolutional Neural Network (CNN) dan ELMo Embeddings

Anak Agung Istri Intan Permata Sari^{a1}, I Ketut Gede Suhartana^{a2}, I Gede Surya Rahayuda^{a3},
I Made Widhi Wirawan^{a4}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas
Udayana Kuta Selatan, Badung, Bali, Indonesia

¹agungintan63@gmail.com

²ikg.suhartana@unud.ac.id

³igedesuryarahayuda@unud.ac.id

⁴made_widhi@unud.ac.id

Abstract

The rapid development of social media in the digital era has led to various negative impacts. One of them is cyberbullying, which can cause serious psychological effects on victims. This study aims to analyze the best combination of hyperparameter tuning for a cyberbullying classification model using a Convolutional Neural Network (CNN) with Embedding from Language Model (ELMo) word representation, as well as to examine the effect of using stemming in the preprocessing stage on accuracy. The data used consists of English-language cyberbullying data obtained from the Kaggle website, with five classification categories: not_cyberbullying, gender, religion, age, and ethnicity. The results show that hyperparameter tuning produces the same best configuration for both conditions—without stemming and with stemming—namely 256 filters and a learning rate of 0.0001. The testing accuracy achieved is 0.8957 for the model without stemming and 0.9017 for the model with stemming. The testing loss is 0.2891 for the model without stemming and 0.2923 for the model with stemming. The application of stemming does not provide a significant improvement in accuracy and even increases the model loss, considering that ELMo as a contextual embedding is already capable of capturing morphological variations of words contextually, making the contribution of stemming limited.

Keywords: Cyberbullying, Text Classification, Convolutional Neural Network, ELMo Embeddings, Stemming, Hyperparameter Tuning.

1. Pendahuluan

Peningkatan penggunaan media sosial yang diiringi dengan kebebasan berpendapat semakin mengkhawatirkan di era digital saat ini karena memicu berbagai dampak negatif, salah satunya cyberbullying. Berdasarkan survei keamanan daring oleh [1] dengan 14.797 responden dari 15 negara, sebanyak 23% pengguna pernah mengalami cyberbullying, meningkat 2% dari tahun sebelumnya. Salah satu kasus yang banyak disorot adalah yang dialami oleh Amanda Todd [2], di mana korban mengalami pelecehan seksual secara daring yang berdampak serius pada kondisi psikologis hingga berujung pada keputusan mengakhiri hidup. Dampak cyberbullying tidak hanya menimbulkan rasa tidak nyaman dalam penggunaan media sosial, tetapi juga memicu gangguan kesehatan mental seperti kecemasan berlebih, kesedihan mendalam, hingga serangan panik. Oleh karena itu, diperlukan suatu pendekatan yang mampu mengidentifikasi dan mencegah tindakan cyberbullying secara efektif melalui proses klasifikasi teks.

Namun, pendekatan konvensional seperti klasifikasi manual memiliki keterbatasan dalam hal kecepatan, konsistensi, dan skalabilitas, terutama seiring meningkatnya volume data teks di media sosial. Berbagai penelitian telah mengusulkan metode klasifikasi otomatis, salah satunya menggunakan *Convolutional Neural Network* (CNN) yang terbukti mampu mengekstraksi fitur penting secara otomatis dan efektif dalam tugas klasifikasi teks. Selain itu, penggunaan *Embedding from Language Model* (ELMo) memungkinkan representasi kata yang kontekstual sehingga mampu menangani polisemi kata dengan lebih baik.

Kombinasi model *Convolutional Neural Network* (CNN) dengan *Embedding from Language Model* (ELMo) memungkinkan penangkapan makna kontekstual dan pola penting secara efektif sehingga menghasilkan kinerja dengan akurasi yang baik. Penelitian yang dilakukan oleh [3] membandingkan kombinasi model yang menghasilkan performa terbaik dalam klasifikasi aspek dan sentimen review restoran di Indonesia. Model yang dibandingkan yaitu BERT-CNN, ELMo-CNN, dan Word2Vec-CNN. Hasil akurasi terbaik pada klasifikasi aspek adalah model kombinasi ELMo-CNN dengan nilai *precision* 88%, *recall* 84%, dan *f1-score* 86%. Kekurangan penelitian di atas yaitu tidak membahas secara detail terkait model klasifikasi yang digunakan, termasuk ELMo-CNN dan *hyperparameter tuning* yang digunakan.

Berdasarkan hal tersebut, penelitian ini berfokus pada penerapan klasifikasi *cyberbullying* menggunakan kombinasi ELMo dan CNN pada data teks media sosial. Selain itu, penelitian ini juga mengeksplorasi kombinasi *hyperparameter tuning* yang optimal untuk meningkatkan kinerja model. Penelitian ini juga membandingkan pengaruh penggunaan *stemming* pada tahap *preprocessing* terhadap performa model, mengingat *stemming* merupakan tahapan *preprocessing* yang memberikan pengaruh signifikan pada struktur data teks. Selain itu, Penelitian [4] yang merupakan publikasi asli yang memperkenalkan ELMo tidak membahas secara spesifik tahapan *preprocessing* yang direkomendasikan sehingga penelitian ini berupaya membuktikan secara empiris pengaruh *stemming* terhadap performa model. Dengan demikian, diharapkan penelitian ini tidak hanya mampu menghasilkan model klasifikasi yang lebih akurat dan efisien, tetapi juga berkontribusi dalam upaya pencegahan *cyberbullying* melalui identifikasi dini terhadap pola-pola ujaran yang bersifat personal dan diskriminatif, serta dapat dimanfaatkan sebagai media edukasi bagi masyarakat dalam mengenali dan mengantisipasi *cyberbullying* di lingkungan digital.

2. Metodologi Penelitian

2.1 Preprocessing

Teks Preprocessing adalah serangkaian langkah untuk memproses data mentah agar dapat digunakan dalam analisis *natural language preprocessing* (NLP). *Preprocessing* merupakan tahapan awal sebelum kumpulan dokumen di klusterisasi yang bertujuan untuk menyeragamkan dataset agar mudah diolah [3]. Tahapan *preprocessing* dalam data teks meliputi *cleaning data* dan *stemming*. Data yang dihasilkan setelah tahapan *preprocessing* yaitu data tanpa *stemming* dan data *stemming*. Pada kondisi kedua data tersebut dilakukan pembersihan data seperti penghapusan duplikat dan penanganan outlier data. Lalu dilanjutkan dengan penghapusan label *other_cyberbullying*

Tahap selanjutnya yaitu random undersampling dengan target 3.000 per kelas sehingga total keseluruhan yaitu 15.000 data. Undersampling diawali dengan membentuk indeks yang merepresentasikan data terpilih. Lalu, *indeks undersampling* diterapkan secara konsisten pada kedua data, yaitu dataset tanpa *stemming* dan dataset dengan *stemming* sehingga keselarasan dan kesesuaian data tetap terjaga.

2.2 Embedding from Language Model (ELMo)

Embedding from Language Model (ELMo) merupakan *contextualized word representations* yang dikembangkan untuk mengatasi pemodelan kata yang kompleks, seperti semantik, dan penggunaan konteks linguistik atau *polysemy*. Penambahan representasi ELMo secara signifikan meningkatkan hasil persentase di setiap kasus, termasuk pengurangan kesalahan hingga 20% [4]. Model representasi ELMo mengadopsi LSTM dua arah atau BiLSTM untuk mempelajari hubungan konteks antar rangkaian kata [5]. Pendekatan dua arah dapat memastikan bahwa konteks sebelumnya dan konteks yang akan datang dipertimbangkan untuk setiap token. Untuk setiap kalimat dengan n token, $W = w_1 +$

$w_2 + \dots + w_n$, forward LSTM akan menghitung probabilitas setiap token w_i berdasarkan urutan sebelumnya ($w_1 + w_2 + \dots + w_{i-1}$) seperti persamaan dibawah ini :

$$p(w_1 + w_2 + \dots + w_n) = \prod_{i=1}^n p(w_i | w_1 + w_2 + \dots + w_{i-1}) \quad (1)$$

Sama halnya dengan *forward* LSTM, *backward* LSTM menggunakan perhitungan yang serupa tetapi arah terbalik. Seperti pada persamaan di bawah ini :

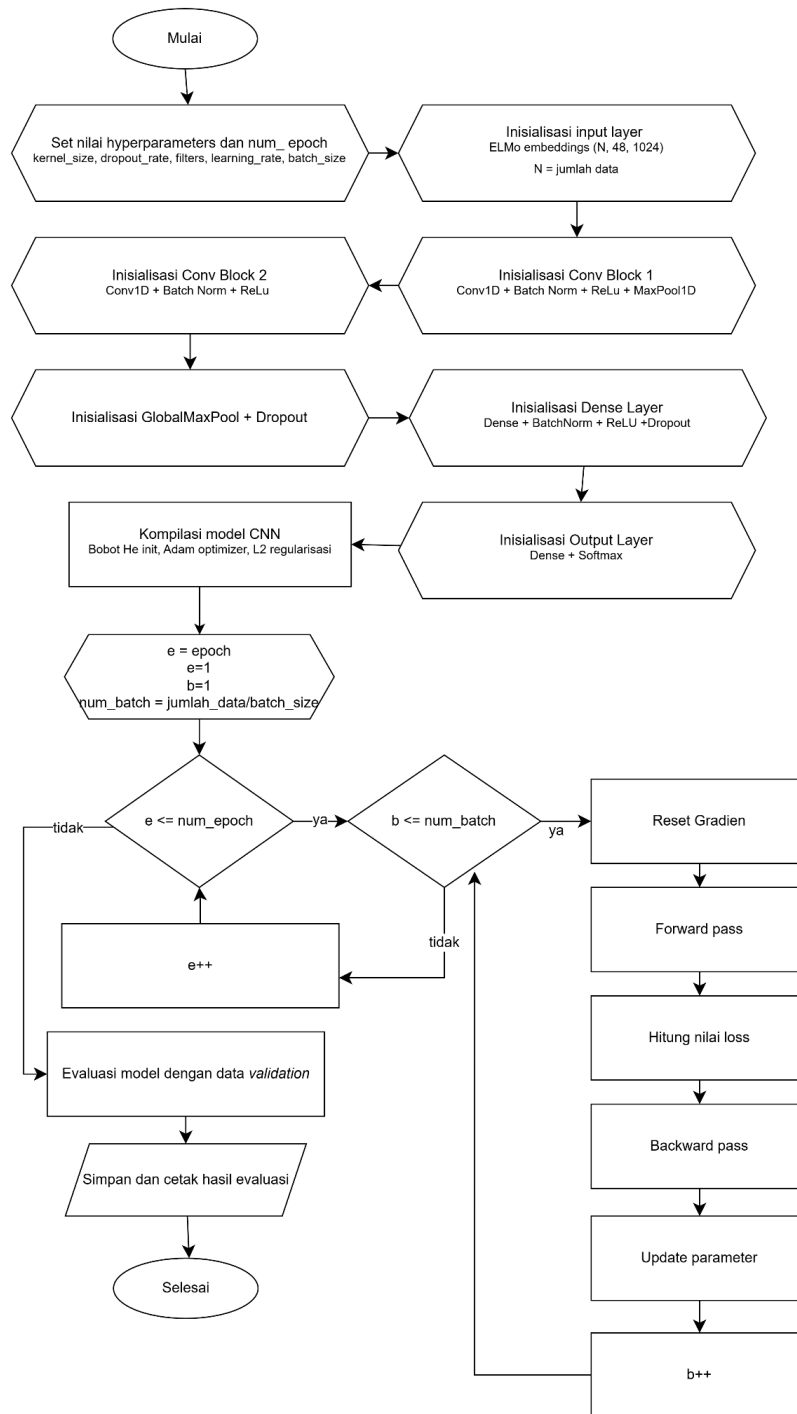
$$p(w_1 + w_2 + \dots + w_n) = \prod_{i=1}^n p(w_i | w_{i+1} + w_{i+2} + \dots + w_n) \quad (2)$$

Dalam pemodelannya, ELMo memiliki tiga lapisan, L1, L2, L3, yang menerima teks asli kemudian mengekstraknya menjadi vektor. ELMo embeddings digunakan sebagai *pre-trained* model pada penelitian ini.

2.3 Convolutional Neural Network (CNN)

CNN adalah jenis arsitektur jaringan saraf yang bisa digunakan untuk pengolahan data, termasuk klasifikasi teks. Pada dasarnya CNN memiliki arsitektur yang terdiri dari *convolutional layers*, *pooling layers*, dan *fully connected layers* [6]. Dalam penerapannya pada data sekuensial seperti *teks*, arsitektur yang digunakan adalah CNN satu dimensi (CNN-1D). CNN-1D salah satu pendekatan yang efektif dalam klasifikasi teks karena mampu memproses data dengan baik serta mengidentifikasi pola dan fitur yang penting [7]. *Convolutional layers* menerapkan filter untuk memindai vektor pada data yang sudah melewati proses *embedding*. Prosesnya melibatkan pengambilan matriks kecil angka (kernel atau filter) dan menggesernya di atas matriks paragraf, lalu mengubah berdasarkan nilai dari filter tersebut. *Convolutional layers* berfungsi untuk mengekstrak makna mendalam dari kalimat yang telah di *train* sebelumnya. lapisan ini membantu memahami dan menangkap konteks serta hubungan antar kata dalam kalimat untuk digunakan pada *layers* selanjutnya. Output dari *convolutional layers* yaitu *feature maps* yang berupa matriks baru. Operasi *convolutional layers* menggunakan filter $W \in R^{h \times k}$, yang diterapkan pada jendela yang terdiri dari h kata untuk menghasilkan fitur baru.

Pooling layers berfungsi untuk mengurangi kompleksitas dimensi [6]. Pengurangan kompleksitas dimensi bertujuan untuk mengurangi *overfitting* dan meningkatkan efisiensi komputasi dengan mengurangi jumlah parameter dan komputasi yang diperlukan pada *layers* berikutnya. Salah satu jenis *pooling layers* yang sering digunakan yaitu *Max Pooling*. *Max pooling* memilih elemen maksimum dari *feature maps*. Output dari *max pooling* yaitu *feature maps* yang mengandung fitur-fitur paling menonjol dari *feature* pada *maps layers* sebelumnya. *Feature maps* yang dihasilkan dari *pooling layers* dimasukkan ke *fully connected layers* untuk proses klasifikasi berdasarkan labelnya. *Layers* terakhir dalam model CNN yaitu *fully connected layers* yang menghasilkan vektor dimensi k . k merupakan jumlah label yang diproses dengan vektor yang berisikan probabilitas untuk setiap label dari teks yang diklasifikasikan. *Fully connected layers* menggunakan fungsi *softmax* yang menyediakan luaran klasifikasi. Disamping itu, Lapisan *dropout* juga digunakan untuk meningkatkan generalisasi model sehingga mencegah *overfitting*. Adapun implementasi CNN-1D pada penelitian ini sebagai berikut :



Gambar 1. Flowchart CNN

Adapun tabel yang menunjukkan detail arsitektur CNN yaitu sebagai berikut :

Tabel 1. Arsitektur CNN

Nama Layer	Output	Params
Conv1D	[B 48, <i>filters</i>]	kernel_size, elmo_dim, filters
Batch Normalization	[B, 48, <i>filters</i>]	2 x <i>filters</i>
ReLU	[B, 48, <i>filters</i>]	-
MaxPooling1D	[B, 24, <i>filters</i>]	-
Conv1D	[B, 24, <i>filters</i> //2]	kernel_size, elmo_dim, <i>filters</i> //2
Batch Normalization	[B, 24, <i>filters</i> //2]	2 x <i>filters</i> //2
ReLU	[B, 24, <i>filters</i> //2]	-
GlobalMaxPooling1D	[B, <i>filters</i> //2]	-
Dropout	[B, <i>filters</i> //2]	-
Dense	[B, 64]	<i>filters</i> //2 x 64 + 64
Batch Normalization	[B,64]	64 x 2
ReLU	[B,64]	-
Dropout	[B,64]	-
Dense	[B, num_classes]	64 x num_classes + num_classes
Softmax	[B, num_classes]	-

2.4 Hyperparameter Tuning

Hyperparameter tuning dilakukan untuk melatih model dengan melakukan beberapa kombinasi parameter yang telah ditentukan [6]. Pada arsitektur CNN-1D, terdapat beberapa *hyperparameter* utama yang bisa dioptimalkan, antara lain jumlah filter konvolusional (*number of filters*), ukuran kernel (*kernel size*), fungsi aktivasi (*activation function*), *learning rate*, jumlah *epoch*, ukuran batch (*batch size*), serta jumlah unit pada lapisan *fully connected* [8]. Namun pada penelitian ini, *hyperparameter tuning* difokuskan pada *filter* dan *learning rate* karena kedua parameter tersebut memberikan pengaruh signifikan terhadap performa model CNN-1D. Metode yang digunakan dalam *hyperparameter tuning* yaitu *Grid Search*. *Grid search* merupakan metode optimasi hyperparameter yang bekerja dengan melakukan pelatihan model pada seluruh kombinasi parameter yang ada hingga menghasilkan konfigurasi terbaik [9]. Setiap kombinasi akan digunakan untuk melatih model secara terpisah dan performanya dievaluasi dengan *validation loss*. Penelitian yang dilakukan oleh [10] dengan yang menggunakan *grid search* pada model hybrid CNN

2.5 Confusion Matrix

Confusion matrix digunakan dalam mengevaluasi seberapa bagus suatu metode yang digunakan [11]. Matriks ini terdiri dari beberapa komponen utama, yaitu *True Positive* (TP) yang menunjukkan jumlah data berlabel positif dan berhasil diprediksi sebagai positif, *True Negative* (TN) yang merupakan jumlah data berlabel negatif dan diprediksi dengan benar sebagai negatif, *False Positive* (FP) yaitu data berlabel negatif tetapi salah diprediksi sebagai positif, serta *False Negative* (FN) yaitu data berlabel positif namun keliru diprediksi sebagai negatif. Selain itu, terdapat beberapa metrik evaluasi yang umum digunakan diantaranya : *accuracy*, *precision*, *recall*, dan *f1-score*. *Accuracy* digunakan untuk mengukur tingkat ketepatan keseluruhan prediksi model, baik pada kelas positif maupun negatif. *Precision* menggambarkan proporsi prediksi positif yang benar dari seluruh prediksi positif yang dihasilkan model. *Recall* menunjukkan kemampuan model dalam menemukan seluruh data yang benar-benar positif. Sementara itu, *f1-score* merupakan rata-rata harmonis antara *precision* dan *recall* yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut. Berikut ini merupakan rumus dari *accuracy*, *precision*, *recall*, dan *f1-score* :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$f1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

3. Hasil dan Pembahasan

3.1. Pelatihan Model

Pelatihan model klasifikasi CNN dengan data tanpa *stemming* dan data *stemming* dilakukan dengan skenario yang sama. Pelatihan model dilakukan proses *hyperparameter tuning* menggunakan *grid search*. Hyperparameter yang diuji yaitu *filters* dan *learning rate* . Adapun nilai dari parameter yang digunakan dalam proses *hyperparameter tuning* yaitu sebagai berikut :

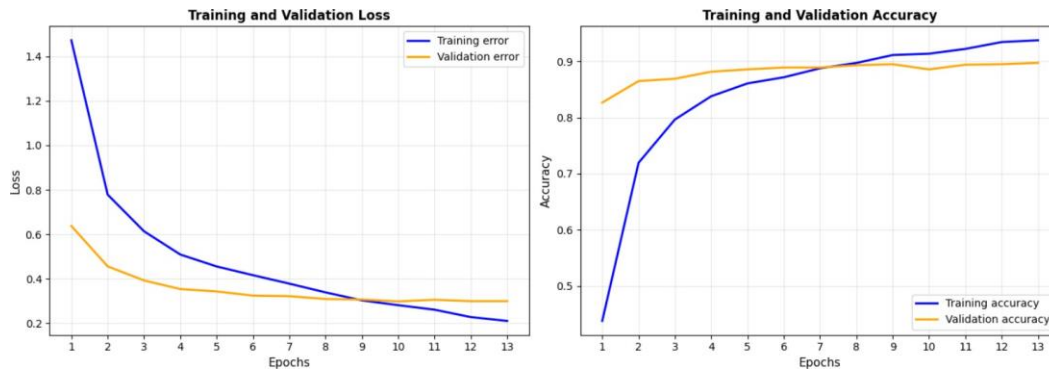
Tabel 2. Konfigurasi Hyperparameter

No	Parameter	Nilai
1	Filters	64, 128, 256
2	Learning Rate	0.0001 ; 0.001 ; 0.01

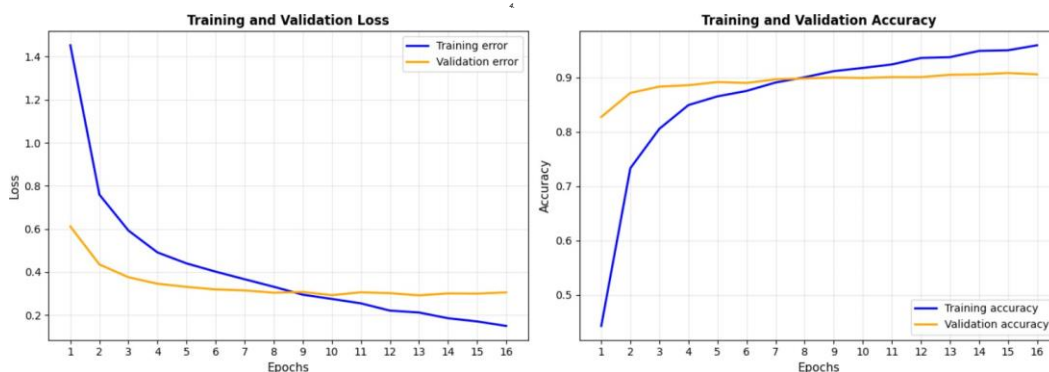
Selain *filters* dan *learning rate*, terdapat hyperparameter lainnya yang bernilai tetap pada tahap pelatihan model CNN yaitu : *kernel size* = 3, *dropout* = 0.5, *batch size* = 32, *epoch* = 30, *early stopping patience* =3, *random seed* = 42. Best hyperparameter diperoleh berdasarkan nilai *validation loss* terendah dari seluruh kombinasi dalam *grid search*. Berdasarkan hasil pelatihan, model tanpa *stemming* dan model *stemming* menghasilkan konfigurasi hyperparameter yang identik yaitu *filters* sebesar 256 dan *learning rate* sebesar 0.0001.

3.2. Evaluasi Model

Evaluasi dilakukan pada model tanpa *stemming* dan model *stemming* yang telah dilatih sebelumnya. Adapun evaluasi performa model best hyperparameter pada pelatihan yaitu sebagai berikut :



Gambar 2. Visualisasi Performa Model tanpa *Stemming*



Gambar 3. Visualisasi Performa Model *Stemming*

Berdasarkan gambar, diperoleh nilai *training loss* dan *validation loss* pada kedua model mengalami penurunan yang cukup tajam pada beberapa *epoch* awal, kemudian cenderung menunjukkan kestabilan seiring bertambahnya jumlah *epoch*. Kondisi ini mengindikasikan bahwa model mampu mempelajari pola dari data pelatihan dengan baik tanpa menunjukkan peningkatan kesalahan yang signifikan. Pada model tanpa *stemming*, *validation accuracy* mencapai titik terbaik sebesar 89,75% pada *epoch* ke-13. Sedangkan model *stemming*, *validation accuracy* mencapai titik terbaik sebesar 90,83% pada *epoch* ke-16. Terlihat bahwa model *stemming* lebih unggul dari segi *validation accuracy* sebesar 1,08%. Perbedaan nilai ini tidak begitu signifikan. Lalu, model tanpa *stemming* mencapai *training accuracy* terbaik sebesar 89,75% ; sedangkan model *stemming* mencapai *training accuracy* terbaik sebesar 95,94%. Model *stemming* memang menghasilkan *train accuracy* yang lebih tinggi. Namun *overfitting gap* model *stemming* (5,11%) lebih besar dibandingkan model tanpa *stemming* (4,02%). *Overfitting gap* digunakan untuk menilai kemampuan generalisasi model yang dihitung dari selisih *training accuracy* terbaik dan *validation accuracy* terbaik. Apabila dilihat dari *epoch* yang diperlukan oleh masing-masing model untuk mencapai performa terbaik, model tanpa *stemming* cenderung lebih cepat konvergen dibandingkan model *stemming*, yakni pada *epoch* ke-13 berbanding *epoch* ke-16. Hal ini mengindikasikan bahwa proses *stemming* yang mengubah bentuk kata ke bentuk dasarnya justru mempersulit model dalam menemukan representasi optimal lebih awal.

Selanjutnya, dilakukan evaluasi pengujian model dengan *test accuracy* dan *test loss* serta *metrics* evaluasi. Berikut detail perbandingan pengujian kedua model :

Tabel 3. Perbandingan *Test Loss* dan *Test Accuracy*

	<i>Test Loss</i>	<i>Test Accuracy</i>
Model tanpa <i>stemming</i>	0.2891	0.8957
Model <i>stemming</i>	0.2923	0.9017

Berdasarkan tabel di atas, dilihat *test accuracy* model *stemming* sedikit lebih unggul dengan selisih

sebesar 0.0042. Namun di sisi lain, model tanpa *stemming* justru menghasilkan test *loss* yang lebih rendah dibandingkan model *stemming* dengan selisih 0.0032. Test *loss* menggambarkan seberapa besar kesalahan prediksi model terhadap data *testing*. Selisih performa testing pada kedua model sangat kecil yang menunjukkan bahwa penerapan *stemming* tidak memberikan dampak yang berarti terhadap kemampuan model dalam mengklasifikasikan data *testing*.

Evaluasi dan perbandingan model tanpa *stemming* dan model *stemming* juga dilakukan pada *metrics* evaluasi yaitu *precision*, *recall*, *f1-score*, dan *accuracy*.

Tabel 4 Evaluasi Model tanpa *Stemming*

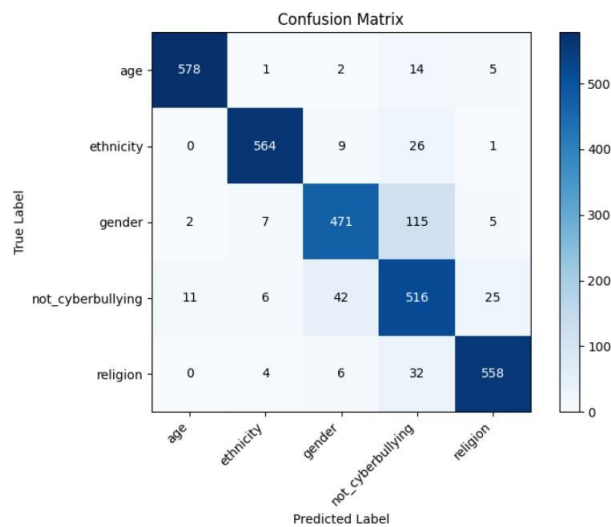
Kelas	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>accuracy</i>
<i>age</i>	0.98	0.96	0.97	0.90
<i>ethnicity</i>	0.97	0.94	0.95	
<i>gender</i>	0.89	0.79	0.83	
<i>not_cyberbullying</i>	0.73	0.86	0.79	
<i>religion</i>	0.94	0.93	0.93	

Tabel 5. Evaluasi Model *Stemming*

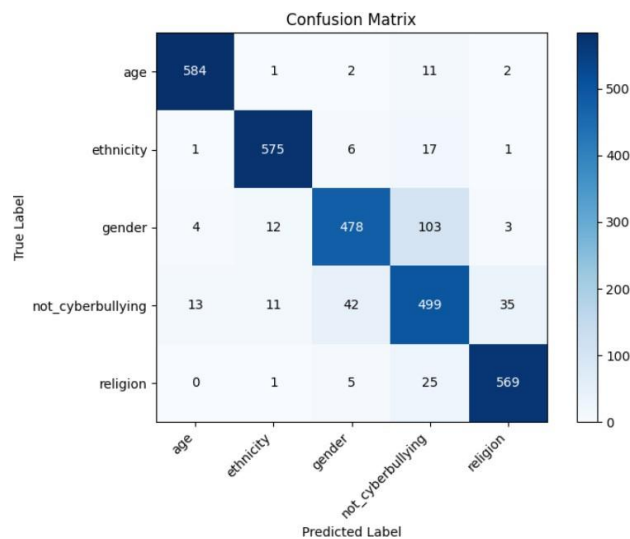
Kelas	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>accuracy</i>
<i>age</i>	0.97	0.97	0.97	0.90
<i>ethnicity</i>	0.96	0.96	0.96	
<i>gender</i>	0.90	0.80	0.84	
<i>not_cyberbullying</i>	0.76	0.83	0.80	
<i>religion</i>	0.93	0.95	0.94	

Dari kedua tabel diatas, dapat dilihat bahwa model tanpa *stemming* dan model *stemming* menunjukkan pola performa yang serupa pada setiap kelasnya. Kelas *age* diikuti kelas *ethnicity* secara konsisten memperoleh nilai *precision*, *recall*, dan *f1-score* tertinggi pada kedua model. Hal ini mengindikasikan bahwa kedua kelas tersebut memiliki karakteristik fitur yang paling mudah dikenali. Sebaliknya, kelas *not_cyberbullying* secara konsisten memperoleh nilai terendah pada kedua model yang menunjukkan bahwa kelas ini paling sulit diklasifikasikan. Secara keseluruhan, kedua model memperoleh nilai *accuracy* yang sama sebesar 0,90.

Terakhir, dilakukan evaluasi terhadap hasil pengujian model dengan *confusion matrix*. Confusion Matrix dari model tanpa *stemming* dan model *stemming* sebagai berikut :



Gambar 4 Confusion Matrix Model tanpa Stemming



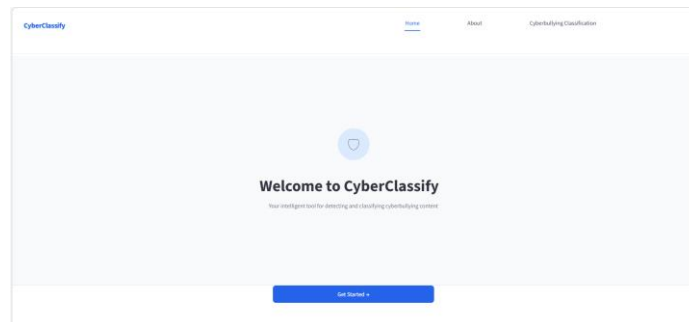
Gambar 5. Confusion Matrix Model Stemming

Dari gambar diatas, terlihat odel *stemming* menunjukkan jumlah prediksi benar yang lebih tinggi pada kelas *age*, *ethnicity*, *religion*, dan *gender* dengan selisih berturut-turut sebesar 6, 11, 11, dan 7 sampel. Sebaliknya, model tanpa *stemming* justru lebih unggul pada kelas *not_cyberbullying* dengan selisih sebesar 17 sampel, yakni 516 sampel berbanding 499 sampel. Selisih performa prediksi dari kedua model tidak terlalu besar jika dibandingkan dengan jumlah data *testing* per label yaitu 600 sampel, yang mana selisih tersebut hanya berkisar antara 1% hingga 2,8% dari total sampel per label. Hal ini menunjukkan bahwa kedua model memiliki kemampuan klasifikasi yang relatif setara secara keseluruhan.

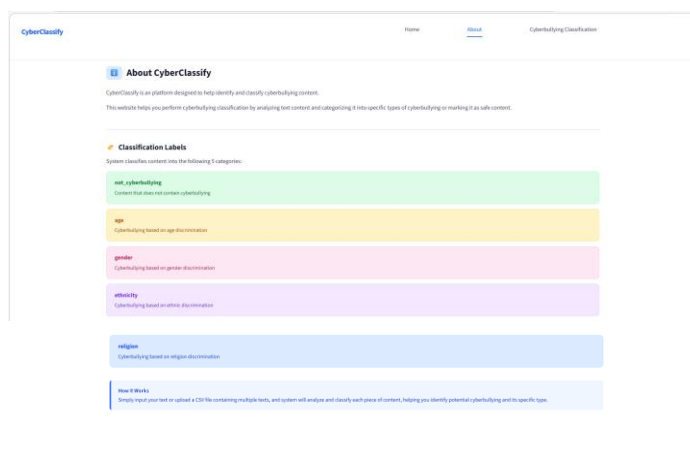
4.1. Implementasi Antarmuka Sistem

Antarmuka sistem klasifikasi *cyberbullying* diimplementasikan menggunakan framework Streamlit. Sistem ini dibangun dengan tiga halaman utama yang dapat diakses melalui navigasi pada bagian atas antarmuka, yaitu halaman *Home*, *About*, dan *Cyberbullying*

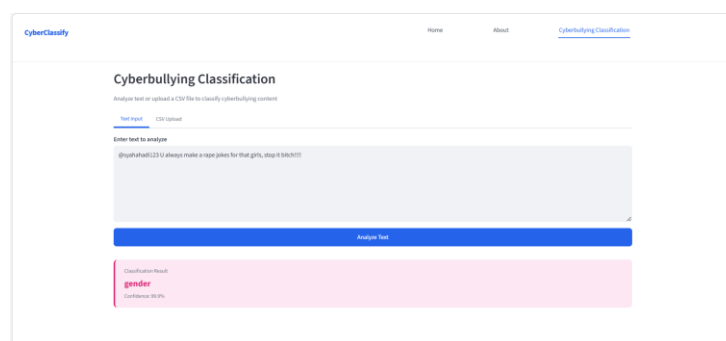
Classification. Halaman *Cyberbullying Classification* merupakan halaman yang digunakan untuk melakukan proses klasifikasi. Halaman ini menyediakan dua mode input yang dapat dipilih melalui tab yaitu *Text Input* dan *CSV Upload*. Pada tab *Text Input*, pengguna dapat mengetikkan teks yang ingin dianalisis ke dalam kolom teks yang tersedia kemudian menekan tombol "*Analyze Text*" untuk memulai proses klasifikasi. Pada tab *CSV Upload*, pengguna dapat mengunggah file *CSV* yang berisi sejumlah teks secara sekaligus. Pengguna juga dapat memilih kolom pada file *.csv* yang ingin dianalisis. Hasil klasifikasi kemudian ditampilkan kepada pengguna beserta *confidence score* dari setiap prediksi yang dihasilkan oleh model CNN *tanpa stemming* yang telah dilatih sebelumnya.



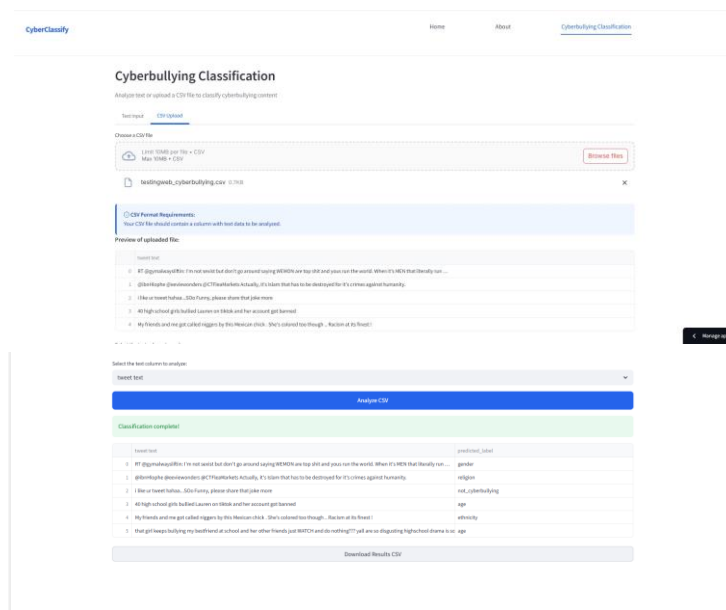
Gambar 6. Tampilan Halaman *Home*



Gambar 7. Tampilan Halaman *About*



Gambar 8. Tampilan Halaman *Cyberbullying Classification Text Input*



Gambar 9. Tampilan Halaman *Cyberbullying Classification CSV Upload*

5. Kesimpulan

Berdasarkan hasil dari penelitian ini, maka dapat ditarik kesimpulan yang dijabarkan sebagai berikut :

- a. Kombinasi *hyperparameter* terbaik untuk model *Convolutional Neural Network (CNN)* pada kedua kondisi *preprocessing* menghasilkan konfigurasi yang identik. Baik pada model tanpa *stemming* maupun model dengan *stemming*, proses *hyperparameter tuning* menghasilkan konfigurasi terbaik yaitu *filters* sebesar 256 dan *learning rate* sebesar 0,0001. Lalu adapun parameter tetap berupa *kernel size* sebesar 3, *dropout rate* sebesar 0,5, dan *batch size* sebesar 32
- b. Penggunaan *stemming* pada *preprocessing* tidak memberikan pengaruh yang berarti terhadap peningkatan performa model klasifikasi *cyberbullying* menggunakan CNN dan ELMo *embeddings*. Model tanpa *stemming* memperoleh *test accuracy* sebesar 0.8957 dan *test loss* sebesar 0.2891. Sedangkan model *stemming* memperoleh nilai *test accuracy* sebesar 0.9017 dan *test loss* sebesar 0.2923 Berdasarkan *metrics* evaluasi menggunakan *precision*, *recall*, dan *f1-score*, kedua model menunjukkan pola performa yang serupa. Kelas *age* diikuti kelas *ethnicity* secara konsisten memperoleh nilai tertinggi pada kedua model, sedangkan kelas *not_cyberbullying* secara konsisten memperoleh nilai terendah pada kedua model. Nilai *accuracy* yang diperoleh oleh kedua model juga sama yaitu 0.90. Selain itu, hasil *confusion matrix* menunjukkan bahwa selisih jumlah prediksi benar antar kedua model pada setiap kelas tergolong kecil berkisar antara sekitar 1% hingga 2,8% dari total data per kelas. Hal ini disebabkan karena ELMo *embeddings* sudah mampu memahami konteks dan variasi bentuk kata sehingga keberadaan *stemming* menjadi kurang berpengaruh terhadap hasil akhir model.

References

- [1] Microsoft, "Digital Safety" Global Online Safety Survey Results, 2025. [Online]. Available: <https://www.microsoft.com/en-us/digitalsafety/research/global-online-safety-survey>. [Accessed: Apr. 10, 2026].
- [2] E. E. Palilingan, R. D. O. Hutabarat and R. K. Pramigoro, "Upaya Pencegahan Untuk Mengurangi Kasus Cyberbullying Di Kalangan Remaja" TUTURAN: Jurnal Ilmu Komunikasi, Sosial dan Humaniora, vol. 2, no. 2, pp. 185–193, 2024.
- [3] A. Amalia, M. S. Lydia, S. D. Fadilla and M. Huda, "Perbandingan Metode Klaster dan Preprocessing Untuk Dokumen Berbahasa Indonesia" Jurnal Rekayasa Elektrika, vol. 14, no. 1, pp. 35–42, 2018.
- [4] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep Contextualized Word Representations" in Proceedings of the 2018 Conference of the North

American Chapter of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2018, pp. 2227–2237.

- [5] Z. Huang and W. Zhao, "Combination of ELMO Representation and CNN Approaches to Enhance Service Discovery" *IEEE Access*, vol. 8, pp. 130782–130796, 2020.
- [6] S. Z. Adhari, "Identifikasi Ujaran Kebencian pada Twitter Menggunakan Metode Convolutional Neural Network (CNN)" *e-Proceeding of Engineering*, vol. 10, pp. -, n.d.
- [7] C. R. S. Hati and H. Sulistiani, "Implementation of CNN Algorithm for Indonesian Hoax News Detection on Online News Portals" *Journal of Applied Informatics and Computing*, vol. 9, no. 3, pp. 765–774, 2025.
- [8] T. N. Tran, "Grid Search of Convolutional Neural Network model in the case of load forecasting" *Archives of Electrical Engineering*, pp. 25–30, 2021.
- [9] Z. Kirori, "Hyper-parameter optimization: towards practical sentiment analysis using a Convolutional Neural Network (CNN)" *Research Journal of Computer and Information Technology Sciences*, vol. 7, no. 1, pp. 1–5, 2019.
- [10] J. Tembhurne, A. Agrawal and K. Lakhotia, "COVID-19 Twitter Sentiment Classification Using Hybrid Deep Learning Model Based on Grid Search Methodology" Cornell University, 2024.
- [11] S. Simanjuntak, S. Paabanan, S. Berutu, S. Sandino and G. C. Setyawan, "Implementasi Metode CNN pada Klasifikasi Sentimen terhadap Pelaksanaan Piala Dunia U-17" *Journal of Engineering and Emerging Technology*, vol. 2, no. 2, pp. 23–32, 2024.