Rekayasa Hijau: Jurnal Teknologi Ramah Lingkungan Volume 9 | Nomor 1
ISSN [e]: 2579-4264 | DOI: https://doi.org/10.26760/jrh.V9i1.58-69 Maret 2025

Klasifikasi *Cyberbullying* Pada *Tweet* Bahasa Sunda Dengan Menggunakan *Hybrid Learning Model*

Anisa Putri Setyaningrum[1], Muhammad Fahmy Nadhif[2]
Program Studi Informatika, Institut Teknologi Nasional, Bandung, Indonesia
Program Studi Informatika, Institut Teknologi Bandung, Bandung, Indonesia
Email: anisaputrisetyaningrum@itenas.ac.id[1], fahmynadhif@gmail.com[2]

Received 15 Januari 2025| Revised 25 Januari 2025| Accepted 2 Februari 2025

ABSTRAK

Cyberbullying dalam bahasa Sunda semakin marak di media sosial, dengan kasus seperti penghinaan fisik, body shaming, dan ancaman yang dapat berdampak negatif pada korban. Namun, deteksi otomatis masih menghadapi tantangan, terutama dalam keterbatasan dataset dan efektivitas metode pemrosesan bahasa alami. Penelitian ini bertujuan untuk mengembangkan sistem deteksi cyberbullying bahasa Sunda menggunakan gabungan model stemming dan hybrid learning. Peneliti menerapkan beberapa model machine learning yaitu random forest dan Support Vector Machine (SVM) serta model deep learning yaitu convolutional neural network-bidirectional long short-term memory (CNN-BiLSTM), CNN, dan BiLSTM. Peneliti melakukan eksperimen untuk mengevaluasi kinerja masing-masing model dengan mengukur akurasi dan F1-score. Berdasarkan hasil penelitian, model hybrid learning memperoleh kinerja terbaik dengan akurasi sebesar 97,3% dan F1-score sebesar 97%. Selain itu, waktu pelatihan pada CNN-BiLSTM lebih cepat dibandingkan dengan model lainnya yaitu sekitar 30 detik per epoch.

***Kata kunci**: Bahasa Sunda, Cyberbullying, Hybrid Learning*

ABSTRACT

Cyberbullying in the Sundanese language is becoming more common on social media, with cases like physical insults, body shaming, and threats that can seriously affect victims. However, detecting it automatically remains challenging, mainly due to limited datasets and the difficulty of processing the language effectively. This study aims to develop a Sundanese cyberbullying detection system using a combination of stemming and hybrid learning models. The researchers applied several machine learning models, namely random forest and Support Vector Machine (SVM), and deep learning models, namely convolutional neural network-bidirectional long short-term memory (CNN-BiLSTM), CNN, and BiLSTM. The researchers conducted experiments to evaluate the performance of each model by measuring the accuracy and F1-score. Based on the results, the hybrid learning model achieved the best performance, with an accuracy of 97.3% and an F1-score of 97%. Besides that, the training time on CNN-BiLSTM is faster than the others which is approximately 30 seconds per epoch.

***Keywords**: Sundanese, Cyberbullying, Hybrid Learning*

1. PENDAHULUAN

X merupakan salah satu platform media sosial terpopuler di dunia, yang membatasi setiap unggahan hingga 280 karakter. Berbagai fitur di X memudahkan pengguna untuk mengekspresikan pikiran mereka secara bebas melalui teks pendek [1]. Namun, berbagai aktivitas di X juga rentan dilakukan oleh pengguna yang menyembunyikan profil aslinya. Beberapa dari mereka menggunakan akun palsu untuk terlibat dalam ujaran kebencian atau perundungan, yang sulit dilakukan di dunia nyata [1]. *Cyberbullying* adalah tindakan melecehkan, mempermalukan, mengancam, atau menyakiti orang lain melalui komputer, telepon seluler, dan perangkat elektronik lainnya [2]. Perundungan siber melalui internet, seperti media sosial, lebih berbahaya daripada perundungan tradisional, karena potensinya untuk menjangkau audiens daring yang tidak terbatas [3]. Menurut survei oleh UNICEF dan Kementerian Komunikasi dan Informasi [4], perundungan siber telah terjadi di Indonesia. Dari 435 remaja (usia 10-19 tahun), 13 persen dari mereka yang menyadari perundungan siber (42 persen dari 435) mengalaminya. Namun, survei ini juga menunjukkan bahwa 58 persen dari 435 remaja tidak memahami *cyberbullying* dan dampak buruknya.

Berdasarkan publikasi Kementerian Pendidikan dan Kebudayaan berjudul "Tinjauan Vitalitas Bahasa Daerah di Indonesia Berdasarkan Data Tahun 2018-2019" [5], terdapat 32,4 juta penutur bahasa Sunda di Indonesia, dengan jumlah suku bangsa Sunda mencapai 36,7 juta jiwa. Hal ini menjadikan bahasa Sunda sebagai bahasa daerah terbesar kedua di Indonesia. Laporan Kementerian Pendidikan dan Kebudayaan (Kemendikbud) tahun 2019 juga menunjukkan bahwa Jawa Barat merupakan provinsi dengan jumlah kasus kekerasan dan bullying tertinggi kedua setelah DKI Jakarta, dengan jumlah kasus kekerasan dan perundungan di Jawa Barat sebanyak 391 kasus [6]. Meskipun *cyberbullying* dalam bahasa Indonesia telah banyak diteliti, penelitian yang berfokus pada deteksi *cyberbullying* dalam bahasa Sunda masih sangat terbatas. Deteksi otomatis *cyberbullying* menjadi tantangan karena banyaknya informasi yang beredar di media sosial, sehingga tidak mungkin dilakukan secara manual [7]. Dalam penelitian ini, peneliti mengembangkan metode deteksi *cyberbullying* berbasis bahasa Sunda menggunakan teknik *machine learning* dan *deep learning*. Penelitian ini penting karena hingga Q2/2022, X memiliki 237,8 juta pengguna aktif, dan tanpa sistem deteksi yang efektif, konten yang mengandung *cyberbullying* dapat dengan mudah menyebar. Dengan mengembangkan sistem klasifikasi berbasis bahasa Sunda, penelitian ini diharapkan dapat membantu dalam mengidentifikasi aktifitas *cyberbullying* di media sosial.

2. METODOLOGI

2.1 Latar Belakang dan Penelitian Terkait

Beberapa penelitian terkait analisis dan deteksi *cyberbullying* telah dilakukan dalam beberapa tahun terakhir. Hani Nurrahmi [2] menggunakan metode *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN) untuk menguji dan mendeteksi teks *cyberbullying* pada data *tweet* berbahasa Indonesia. Hasil penelitian menunjukkan bahwa SVM menghasilkan *F1-Score* tertinggi, yaitu 67%. Meylan Wongkar [8] menganalisis data X kandidat presiden Indonesia 2019 dan membandingkan metode *Naïve Bayes*, SVM, dan KNN menggunakan RapidMiner. Hasil penelitian menunjukkan nilai akurasi *Naïve Bayes* sebesar 75,58%, nilai akurasi SVM sebesar 63,99%, dan nilai akurasi KNN sebesar 73,34%. Maryem Rhanoui [9] menggunakan model *hybrid Convolutional Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM) untuk mengklasifikasikan sentimen dokumen, dan memperoleh akurasi yang baik, yaitu 90,66%. Wang Yue [10] melakukan analisis sentimen menggunakan Word2Vec dan jaringan syaraf tiruan *hybrid* CNN dan Bi-LSTM, memperoleh hasil akurasi yang baik sebesar 91,48%. Dharma [11] dalam penelitiannya melakukan evaluasi kinerja terhadap tiga *word embedding* (*Word2Vec*, *GloVe*, dan *FastText*), menemukan bahwa kinerja *FastText* lebih unggul, mencapai hasil akurasi sebesar 97,2%. Berdasarkan fakta tersebut, penelitian ini akan dilakukan untuk mengklasifikasikan *cyberbullying* pada *tweet* bahasa Sunda dari pengguna X menggunakan model *hybrid learning*. Penelitian ini berfokus pada deteksi *cyberbullying* menggunakan

machine learning dengan menerapkan algoritma *Random Forest* dan SVM, serta model *hybrid* CNN dan Bi-LSTM untuk klasifikasi teks skala besar. Teks dikategorikan ke dalam dua kelas: *cyberbullying* dan bukan *cyberbullying*, dengan karakteristik yang berbeda. Teks *cyberbullying* cenderung mengandung kata-kata kasar, hinaan, sarkasme, atau ancaman, sementara teks *nonbullying* lebih bersifat netral, informatif, atau kritik yang konstruktif. Tantangan utama dalam klasifikasi ini adalah keberagaman bahasa, termasuk penggunaan *code-mixing* antara bahasa Sunda dan Indonesia, serta variasi morfologi bahasa Sunda yang kompleks. Untuk meningkatkan akurasi deteksi, data akan melalui *pre-processing* seperti tokenisasi, *stopword removal*, *stemming*, dan *text cleaning* guna menghilangkan karakter tidak relevan. Dengan memahami pola linguistik ini, model diharapkan mampu membedakan teks *bully* dan *nonbully* secara lebih akurat dalam konteks bahasa daerah.

2.2 Dataset dan Metodologi

Bab ini memberikan gambaran umum tentang kumpulan data yang digunakan dan metodologi eksperimen yang dilakukan dalam penelitian ini. Untuk memberikan pemahaman yang jelas kepada pembaca tentang kumpulan data dan metodologi yang digunakan dalam penelitian ini.

A. Dataset

Dataset yang digunakan merupakan hasil *scrapping* data melalui X menggunakan *tool twint* yang dikumpulkan pada bulan Januari hingga Maret 2023. Kemudian dataset tersebut dianotasi oleh pakar yaitu psikolog secara manual untuk menambahkan label pada dataset tersebut. Distribusi *dataset* tersebut adalah 2000 *tweet* yang mengandung *cyberbullying* dan 2000 *tweet* yang tidak mengandung *cyberbullying* dalam bahasa Sunda. Berikut ini adalah tabel I distribusi *dataset cyberbullying* dalam bahasa Sunda.

Tabel 1. Distribusi Dataset *Cyberbullying*

Label	Distribusi
Cyberbullying	2000
Not *Cyberbullying*	2000
Total	**4000**

Dataset tersebut memiliki total 4000 *tweet* yang dibagi menjadi 2 bagian yaitu data latih sebesar 75% (3000 *tweet*) dan data pengujian sebesar 25% (1000 *tweet*). Label *cyberbullying* mencakup teks yang mengandung unsur penghinaan, ancaman, pelecehan, ujaran kebencian yang ditunjukkan pada suatu kelompok atau individu. Label *not cyberbullying* tidak mengandung unsur penghinaan, pelecehan, atau ancaman. Sampel dataset yang sudah dilabeli oleh pakar untuk masing-masing label dapat dilihat pada Tabel 2.

Tabel 2. Sampel Dataset *Cyberbullying*

Label	Sampel Dataset
Cyberbullying	"@ChelseaFC lebih better nonton persib maen anjing dari pada kandang babi, pelatih bapuk, @RLC komo deui sia anjing anak bupati london, sampah sia"
Not *Cyberbullying*	"beres futsal hareudang carepel, mandi meh teu merenah, beres mandi ngadon beberes kamar anying belegug jadi hareudang jeung carepel deui, emang pinter pisan aing"

B. Metodologi

Untuk mengklasifikasikan *cyberbullying* dalam bahasa Sunda, peneliti menggunakan TF-IDF untuk machine learning. Algoritma *machine learning* yang digunakan dalam penelitian ini adalah *Random Forest* dan *Support Vector Machine* (SVM).

1) TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) digunakan untuk mengidentifikasi kata-kata terpenting dalam dokumen atau korpus untuk setiap label. Cara kerjanya adalah dengan memberikan bobot yang lebih tinggi untuk kata-kata unik, sementara kata-kata umum di seluruh korpus menerima bobot yang lebih rendah [12]. Keuntungan menggunakan TF-IDF adalah menghasilkan representasi berdimensi rendah, di mana setiap dokumen direpresentasikan oleh vektor dengan ukuran yang setara dengan kosakata [13], yang mengurangi risiko *over-fitting*. Skor TF-IDF untuk term t dalam dokumen d, *term frequency* (TF) kata dalam dokumen dikalikan dengan *inverse document frequency* (IDF) kata di seluruh korpus. Rumus untuk menghitung skor TFIDF adalah:

$$\text{TF-IDF} = \text{TF x IDF} \quad (1)$$

Setiap berkas memiliki TF (*Term Frequency*) yang dihitung dengan membagi kata yang muncul dengan jumlah kemunculannya dalam berkas dengan jumlah total kata. IDF (*Inverse Data Frequency*) dihitung dengan rumus berikut:

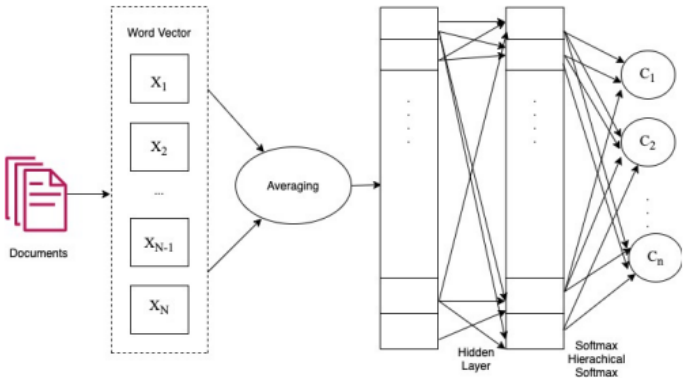
$$f_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \quad (2)$$

Melalui IDF, bobot/frekuensi suatu kata dapat ditentukan dengan logaritma jumlah *file* dibagi dengan jumlah *file* yang mengandung kata "a". Terakhir, TF-IDF dapat dihitung dengan cara mengalikan TF dengan IDF menggunakan persamaan berikut:

$$idf_{i,j} = \log \frac{N}{df_i} \quad (3)$$

2) FastText

Fasttext dikembangkan oleh tim Riset AI Facebook yang digunakan untuk mempelajari representasi kalimat dan kata secara efisien [14]. Tidak seperti *Word2vec* yang menggunakan representasi tingkat kata yang memperlakukan setiap kata sebagai unit terkecil [15], *Fasttext* menggunakan tingkat karakter untuk merepresentasikan kata menjadi vektor. Oleh karena itu, unit terkecil bukanlah kata tetapi karakter. Arsitektur *Fasttext* ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur *Fasttext*

Matriks bobot pertama A digunakan sebagai tabel pencarian untuk vektor kata dari kata-kata dalam kosakata. Dokumen direpresentasikan sebagai kantong kata, di mana setiap kata dalam dokumen dipetakan ke vektor yang sesuai dari matriks bobot A. N-gram vektor kata kemudian dihitung dan dirata-ratakan untuk menghasilkan penyematan kalimat untuk setiap dokumen. Setelah penyematan kalimat dihasilkan untuk semua dokumen dalam korpus, penyematan tersebut selanjutnya dirata-ratakan dan dimasukkan ke pengklasifikasi linier untuk melakukan klasifikasi dokumen.

Untuk setiap kalimat dalam satu set N dokumen, *FastText* menghasilkan penyematan kalimat dengan mengambil rata-rata vektor n-gram yang muncul dalam kalimat. Penyematan kalimat kemudian digunakan untuk memprediksi label kelas dokumen

menggunakan fungsi softmax atau softmax hierarkis f. Fungsi *softmax* menghitung distribusi probabilitas atas kelas-kelas yang telah ditentukan sebelumnya. *FastText* meminimalkan *log-likelihood* negatif atas kelas-kelas untuk semua N dokumen dalam korpus:

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log\left(f(BAx_n)\right) \quad (4)$$

Di mana x_n adalah kumpulan fitur dari dokumen N, label kelas didefinisikan oleh yn, A dan B ditimbang dari matriks. *FastText* biasanya dilatih menggunakan *stochastic gradient descent* (SGD), yang memperbarui parameter model berdasarkan kumpulan data kecil. *Learning rate* menurun secara linier selama proses pelatihan, yang berarti bahwa *Learning rate* menurun seiring dengan peningkatan jumlah iterasi [16].

3) Random Forest

Random Forest adalah algoritma *machine learning* yang dapat digunakan untuk tugas klasifikasi dan regresi [17]. *Random Forest* termasuk dalam keluarga metode pembelajaran *ensemble,* yang berarti menggabungkan beberapa model untuk membuat prediksi yang lebih akurat. *Random Forest* bekerja dengan membuat beberapa pohon keputusan, di mana setiap pohon dibangun di atas subset acak dari data dan fitur. Selama pelatihan, algoritma secara acak memilih subset fitur di setiap node untuk membagi data, yang membantu mengurangi korelasi antara pohon dan mencegah overfitting [18].

Saat membuat prediksi, setiap pohon di *forest* secara independen mengklasifikasikan titik data input atau memprediksi nilainya untuk masalah regresi. Prediksi akhir kemudian dibuat dengan mengambil rata-rata (untuk regresi) atau suara mayoritas (untuk klasifikasi) dari prediksi dari semua pohon. *Random Forest* dikenal karena akurasinya yang tinggi dan ketahanannya terhadap data yang bising, nilai yang hilang, dan *outlier*. Random Forest juga menyediakan ukuran pentingnya fitur, yang dapat membantu pemilihan fitur dan interpretasi model.

4) *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan salah satu metode dalam *supervised learning* yang digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*) [19]. Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik klasifikasi lainnya. SVM juga dapat menangani permasalahan klasifikasi dan regresi dengan data linear maupun non-linear.

SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. Hyperplane merupakan fungsi yang dapat digunakan untuk memisahkan kelas. Dalam 2-D, fungsi yang digunakan untuk klasifikasi antar kelas disebut garis, sedangkan dalam 3-D, fungsi yang digunakan untuk klasifikasi antar kelas disebut bidang. Begitu pula dengan fungsi yang digunakan untuk klasifikasi dalam ruang berdimensi lebih tinggi disebut *hyperplane*. Persamaan *hyperplane* yang ditunjukkan di bawah ini:

$$w^T x + b = 0 \quad (5)$$

Di mana x merupakan masukan vektor mesin, b merupakan bias, dan w merupakan bobot vektor.

Sedangkan untuk deep leaning, peneliti menggunakan *pre-trained fasttext word embedding* dan *hybrid learning*, yaitu CNN-BiLSTM. Selain itu, kami juga menggunakan *metode deep learning* CNN dan Bi-LSTM sebagai pembanding terhadap metode *hybrid* learning.

5) Bidirectional LSTM

BiLSTM (*Bidirectional Long Short-Term Memory*) adalah jenis arsitektur *Recurrent Neural Network* (RNN) yang mampu memproses data berurutan dalam arah maju dan mundur. Blok penyusun dasar BiLSTM adalah sel LSTM (*Long Short-Term Memory*). Sel LSTM adalah jenis sel RNN yang dirancang untuk mengatasi masalah *vanishing gradient* dan masalah *exploding gradient*, yang dapat terjadi saat melatih RNN tradisional. Sel LSTM memiliki sel memori, yang menyimpan informasi dari waktu ke waktu, dan tiga gerbang

(*input gate*, *forget gate*, dan *output gate*) yang mengatur aliran informasi ke dalam dan keluar dari sel memori. Format detail ditunjukkan di bawah ini.

a. Input gate

$$i_t = \sigma(W_i \,.\, x_i + U_i \,.\, h_{t-1} + b_i) \quad (7)$$

b. Transformation

$$\hat{c}_t = tanh(W_c \,.\, x_t + U_c \,.\, h_{t-1} + b_c) \quad (8)$$

c. State update

$$c_t = i_t \odot \hat{c}_t + f_t \odot c_{t-1} \quad (9)$$

d. Output gate

$$o_t = \sigma(W_o \,.\, x_o + U_o \,.\, h_{t-1} + b_o) \quad (10)$$

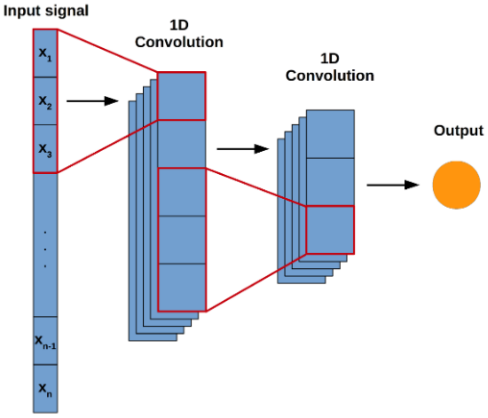
e. Hidden status

$$C_t = o_t \odot \tanh(c_t) \quad (11)$$

6) *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur *deep learning* yang populer yang awalnya dirancang untuk tugas pengenalan dan klasifikasi gambar [20]. Akan tetapi, CNN juga telah berhasil diterapkan pada masalah klasifikasi teks, seperti analisis sentimen, deteksi spam, dan klasifikasi topik [21].

Dalam klasifikasi teks, CNN dapat digunakan untuk mengekstraksi fitur yang relevan dari teks masukan dengan memperlakukan teks sebagai sinyal 1D. CNN menerapkan filter (juga disebut kernel) ke teks masukan, yang berputar di atas teks dan menghasilkan peta fitur. Filter biasanya berukuran kecil dan memanjang hingga tinggi penuh teks masukan.



Gambar 2. 1D CNN Architecture

Misalnya, filter dengan lebar 3 akan mencakup tiga kata berurutan dalam teks. Dalam arsitektur CNN, filter ini secara umum terdiri dari *input layer, convolutional layer, max pooling, fully connected layer*.

a. Input Layer

Pada lapisan input terdapat teks dari *tweet cyberbullying* yang telah diproses sebelumnya dan dikonversi menjadi vektor kata berdimensi 300 menggunakan *word embedding FastText*. Proses ini dilakukan dengan metode *out of vocabulary* sehingga dapat mengakomodasi kosakata yang tidak ditemukan dalam FastText. Dalam satu kalimat terdapat 51 kata, sehingga matriks input akan berukuran 51×100.

b. Convolutional Layer

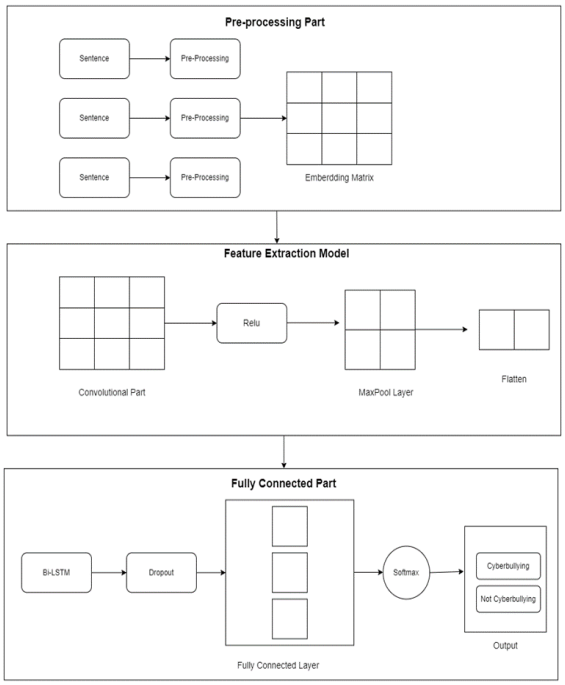
Lapisan konvolusi terdiri dari neuron-neuron yang disusun dalam pola tertentu untuk membentuk filter. Lapisan ini memiliki 128 filter dengan ukuran jendela 5, yang disusun secara vertikal melintasi matriks input. Operasi "titik" dilakukan antara bobot filter dan bobot matriks input, dan setelah itu dilakukan operasi non-linear.

c. Max Pooling

Fungsi aktivasi ReLU menghasilkan peta aktivasi atau *feature map* yang berisi fitur-fitur penting berdimensi rendah di lapisan tersembunyi pertama.

d. Fully Connected Layer

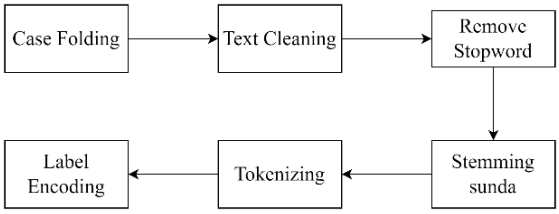
Peta fitur yang telah diubah bentuknya, yang merupakan keluaran dari lapisan tersembunyi sebelumnya, dihubungkan ke lapisan output untuk melakukan klasifikasi. *Softmax* dan *loss function* digunakan dalam lapisan ini karena variabel output biner dikodekan menggunakan *one-hot encoding* yang terdiri dari nilai 0 dan 1.



Gambar 3. Arsitektur *Hybrid* Learning Model

Dalam Gambar 3 dijelaskan bahwa arsitektur model *Hybrid* terdiri dari tiga bagian:

a. ***Preprocessing*:** Pada tahap ini, dilakukan pemrosesan data seperti *case folding*, *text cleaning*, *stopword removal*, tokenisasi, dan *label encoding*. Selanjutnya, teks dikonversi menjadi representasi vektor, yang dapat dilakukan menggunakan *word embedding FastText*. Gambar 4 menunjukkan langkah-langkah *preprocessing* pada *tweet* berbahasa Sunda.



Gambar 4. Preprocessing

b. ***Case Folding*:** Langkah ini melibatkan pengubahan semua teks menjadi huruf kecil atau huruf besar agar teks menjadi seragam. Hal ini penting karena algoritma *machine learning* yang digunakan untuk klasifikasi bersifat *case-sensitive*, yang berarti kata yang sama tetapi ditulis dengan huruf besar dan kecil akan dianggap sebagai dua kata yang berbeda.

c. ***Text Cleaning*:** Langkah ini melibatkan penghapusan karakter atau simbol yang tidak diperlukan dari teks, seperti karakter khusus, tanda baca, URL, dan emotikon.

d. ***Stopword Removal*:** Stopwords adalah kata-kata umum yang tidak memiliki banyak makna dalam suatu kalimat, seperti *"jeung"*, *"sih"*, *"siah"*, *"mah"*, *"tah"*, *"teh"*, *"itu"*, *"ieu"*, *"ka"*, *"di"*, *"ku"*, *"ngan"*, *"nu"*, *"nyah"*, *"oge"*, *"teu"*, *"ti"*, *"wae"*, *"we"*, *"tapi"*, *"sanajan"*, *"salain"*, *"kituna"*, *"sabalikna"*, *"malah"*, dan *"adalah"*

e. ***Stemming***

Stemming adalah proses mengubah setiap kata dalam teks menjadi bentuk dasarnya. Hal ini penting karena berbagai bentuk dari kata yang sama dapat memiliki makna berbeda tetapi dianggap sebagai kata yang berbeda oleh algoritma *machine learning*. Berikut adalah langkah-langkah proses *stemming*:

a) Langkah pertama adalah memeriksa apakah kata input sudah ada dalam kamus kata dasar. Jika ditemukan, kata tersebut dikembalikan sebagai kata dasar.
b) Jika kata tidak ditemukan dalam kamus, langkah selanjutnya adalah menghapus sufiks dari kata input. Sistem akan memeriksa keberadaan sufiks seperti *"ning"*, *"ing"*, *"eun"*, *"keun"*, *"an"*, *"ana"*, *"na"*, *"dua"*, dan *"eta"*. Jika ditemukan, sufiks tersebut akan dihapus.
c) Selanjutnya, sistem menghapus prefiks dari kata input. Sistem akan memeriksa keberadaan prefiks seperti *"barang"*, *"nyang"*, *"silih"*, *"pang"*, *"pada"*, *"para"*, *"per"*, *"ba"*, *"si"*, *"pa"*, *"ti"*, *"ng"*, dan *"mi"*. Jika ditemukan, prefiks tersebut akan dihapus.
d) Langkah berikutnya adalah menghapus infiks, atau disebut *"sisipan"* dalam bahasa Sunda. Sistem akan memeriksa keberadaan infiks seperti *"al"*, *"in"*, *"um"*, dan *"ar"*.
e) Pada setiap langkah, sistem akan memeriksa keberadaan huruf vokal di posisi tertentu dalam kata untuk memastikan bahwa hanya afiks yang sesuai yang dihapus.
f) Akhirnya, kata dasar yang diperoleh dikembalikan sebagai hasil *stemming*.

f. ***Encoding Label*:** Langkah akhir ini mengubah data teks menjadi format numerik agar dapat digunakan oleh algoritma *machine learning*. Hal ini penting karena sebagian besar algoritma *machine learning* memerlukan input data dalam bentuk numerik. Dalam hal ini, *encoding* label menetapkan label numerik untuk setiap kategori dalam masalah klasifikasi.
g. **Bagian Konvolusi:** Pada tahap ini, lapisan konvolusi dan *max pooling* diterapkan untuk melakukan ekstraksi fitur. Tujuannya adalah untuk mendapatkan fitur tingkat tinggi. Hasil dari tahap ini berupa array fitur, yang kemudian menjadi input untuk bagian *fully connected*.
h. **Bagian *Fully Connected*:** Pada tahap ini, lapisan *fully connected* diterapkan pada kalimat yang berisi dugaan *cyberbullying*. Hasil dari tahap ini adalah klasifikasi apakah suatu kalimat mengandung unsur *cyberbullying* atau tidak.

3. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan evaluasi model dan hasil eksperimen. Sistem yang digunakan untuk proses pelatihan adalah *Google Colaboratory* dengan *Python* (3.9.6), *Sklearn* (1.2.2), *Keras* (2.12.0), dan *GPU* 16GB.

3.1 Metrik Evaluasi

Dalam mengevaluasi kinerja model, *F1-Score* dan *Accuracy* digunakan sebagai metrik evaluasi. *F1-Score* mempertimbangkan kedua nilai *Precision* dan *Recall* dalam model, yang juga dianggap sebagai rata-rata harmonik dari *Precision* dan *Recall*, sebagaimana ditunjukkan persamaan (15).

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \quad (15)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \; x \; 100\% \quad (16)$$

Dalam *binary classification*, *True Positive* (TP) adalah jumlah sampel yang terdeteksi dengan benar sebagai kelas target dan sesuai dengan kelas target sebenarnya. *False Positive* (FP) menunjukkan jumlah sampel yang terdeteksi sebagai kelas target tetapi tidak sesuai dengan kelas target sebenarnya. *False Negative* (FN) menunjukkan jumlah sampel yang tidak terdeteksi sebagai kelas target. Metrik ini digunakan untuk mengevaluasi kinerja model klasifikasi [22].

3.2 Eksperimen

Untuk parameter *Random Forest*, kami menggunakan tiga parameter, yaitu *max_depth* yang menunjukkan kedalaman pohon dalam *Random Forest*, *criterion* (Gini dan Entropy), serta *n_estimators* yang menunjukkan jumlah pohon yang dihasilkan oleh *Random Forest*. Tabel 2 berisi parameter-parameter dari *Random Forest*.

Tabel 3. Paramater *Random Forest*

Paramater	***Value***
Max_depth	25,30
critrerion	Entropy,gini
n_estimators	250

Parameter terbaik untuk *Random Forest* adalah *max_depth* 25, *criterion* Entropy, dan *n_estimators* 250. Hasil pelatihan pada Tabel 4. menggunakan algoritma *Random Forest* menunjukkan bahwa model mampu mengklasifikasikan teks dengan baik, dengan *F1-score* sebesar 0.96 untuk kedua kategori serta akurasi keseluruhan mencapai 96%. Evaluasi dilakukan menggunakan metrik *F1-score* dan *Accuracy*, di mana *F1-score* mengukur keseimbangan antara *precision* dan *recall* dalam klasifikasi, sementara Accuracy menunjukkan persentase data yang diklasifikasikan dengan benar. Berdasarkan hasil ini, model *Random Forest* terbukti cukup efektif dalam mendeteksi *cyberbullying* dalam teks berbahasa Sunda.

Tabel 4. Hasil pelatihan menggunakan algoritma Random Forest

Label	***F1-Score***	***Accuracy***
Cyberbullying	0.96	96 %
Not-*cyberbullying*	0.96	96 %
Avg/total	0.96	96 %

Untuk parameter SVM, kami menggunakan tiga parameter, yaitu *kernel*, *C*, *gamma*, dan *degree*. Kernel yang digunakan adalah *linear* dan *RBF* untuk menentukan apakah dataset sesuai. Parameter *C* digunakan untuk mencari nilai *F1-Score* dan akurasi terbaik. Tabel 4 berisi parameter-parameter dari SVM.

Tabel 4. Paramater SVM

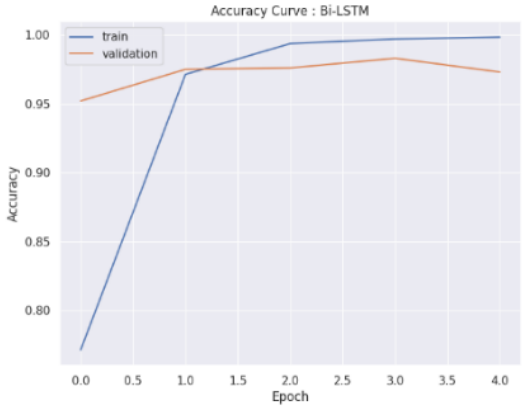
Paramater	***Value***
Kernel	Linear,rbf
C	1,2
gamma	0.01

Parameter terbaik untuk SVM diperoleh dengan menggunakan *kernel* linear dan $C = 2$. Akurasi model terbaik SVM adalah 97%, seperti yang ditunjukkan dalam Tabel 5. Dibandingkan dengan model *Random Forest*, SVM menunjukkan sedikit peningkatan performa, dengan selisih 1% lebih tinggi dalam akurasi dan *F1-score*. Hal ini menunjukkan bahwa SVM lebih unggul dalam mengidentifikasi pola dalam teks berbahasa Sunda, terutama dalam mendeteksi perbedaan antara teks yang mengandung *cyberbullying* dan yang tidak.

Tabel 5. Hasil pelatihan menggunakan algoritma SVM

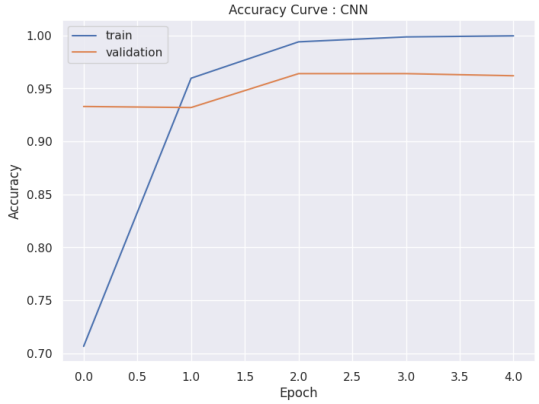
Label	***F1-Score***	***Accuracy***
Cyberbullying	0.97	97 %
Not-*cyberbullying*	0.97	97 %
Avg/total	0.97	97 %

Gambar 5 menunjukkan kurva akurasi model CNN-BiLSTM selama proses pelatihan dan validasi. Akurasi pelatihan meningkat dengan cepat dan mencapai sekitar 97.3%, menandakan bahwa model mampu mengenali pola dalam data pelatihan dengan baik.



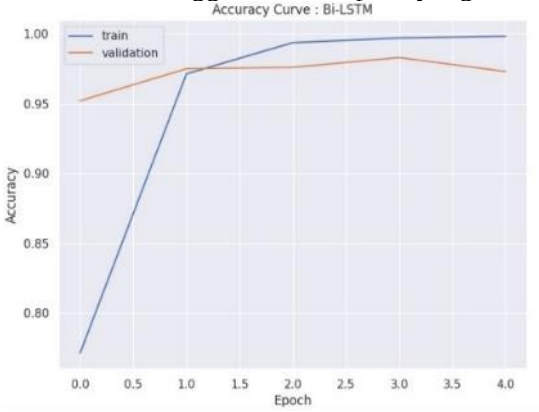
Gambar 5. Akurasi dari CNN-BiLSTM

CNN dapat menghasilkan akurasi hingga 96,2%, seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Akurasi dari CNN

Bi-LSTM dapat menghasilkan akurasi hingga 97,1%, seperti yang ditunjukkan pada Gambar 7.



Gambar 7. Akurasi dari BiLSTM

Tabel 6 menunjukkan perbandingan antara model *machine learning* dan *deep learning*.

Tabel 6. Hasil pelatihan model *machine learning* dan *deep learning*

Model	F1-Score	Accuracy
Random Forest	0.96	96.00%
SVM	0.97	97.00%
CNN	0.96	96.30%
Bi-LSTM	0.97	97.10%
CNN-BiLSTM	**0.97**	**97.30%**

Tabel 6 menunjukkan bahwa model *hybrid learning*, yaitu CNN-BiLSTM, memiliki performa terbaik dibandingkan model lainnya. Hasil performa dari *hybrid learning* menunjukkan bahwa CNN dapat mengekstraksi fitur dengan baik, sementara BiLSTM mampu mempertahankan keterkaitan dan urutan dalam dua arah untuk memahami konteks. Selain itu, waktu pelatihan pada CNN-BiLSTM lebih cepat dibandingkan model lain, yaitu sekitar 30 detik per *epoch*.

Berdasarkan eksperimen untuk pelatihan menggunakan CNN-BiLSTM, CNN, Bi-LSTM menghasilkan akurasi yang cukup baik. Namun, akurasi validasi tetap berada di bawah akurasi pelatihan dan mengalami penurunan pada epoch akhir, yang mengindikasikan kemungkinan *overfitting*. Hal ini dapat disebabkan oleh model yang terlalu menyesuaikan diri dengan data pelatihan sehingga kurang mampu menggeneralisasi data baru, serta perbedaan karakteristik antara dataset pelatihan dan validasi.

4. KESIMPULAN

Sebagai kesimpulan, penelitian ini bertujuan untuk mengembangkan deteksi *cyberbullying* dalam bahasa Sunda menggunakan *stemming* dan model *hybrid* learning. Model yang diterapkan dalam penelitian ini mencakup *machine learning*, yaitu *Random Forest* dan SVM, serta *deep learning*, yaitu CNN-BiLSTM, CNN, dan BiLSTM. Berdasarkan hasil penelitian, model *hybrid learning* menunjukkan performa terbaik dengan akurasi 97,3% dan F1-Score 97%.

Dalam mengatasi *overfitting* yang terjadi pada model dapat diterapkan regularisasi seperti Dropout, menambahkan data *augmentation* untuk meningkatkan variasi teks dalam bahasa Sunda, serta melakukan *hyperparameter tuning* dan *early stopping* guna mencegah pelatihan berlebihan yang dapat menurunkan performa validasi. Selain itu, dapat dilakukan juga eksplorasi lebih lanjut tentang Bi-GRU, yang merupakan pengembangan dari BiLSTM, serta menggabungkannya dengan CNN untuk ekstraksi fitur.

DAFTAR PUSTAKA

[1] J. VAN DIJCK, “Tracing X: The rise of a microblogging platform,” Int. J. Media Cult. Polit., vol. 7, no. 3, pp. 333–348, 2012, doi: 10.1386/macp.7.3.333_1.

[2] H. Nurrahmi and D. Nurjanah, “Indonesian X *Cyberbullying* Detection using Text Classification and User Credibility,” 2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018, vol. 2018-Janua, pp. 543–548, 2018, doi: 10.1109/ICOIACT.2018.8350758.

[3] S. Bauman and M. L. Newman, “Testing assumptions about *cyberbullying*: Perceived distress associated with acts of conventional and cyber bullying,” Psychol. Violence, vol. 3, no. 1, pp. 27–38, 2013, doi: 10.1037/a0029867.

[4] G. Gayatri, “Digital Citizenship Safety among Children and Adolescents in Indonesia.” https://web.kominfo.go.id/sites/default/files/users/12/Kominfo-Presentasi Laporan Hasil Penelitian - Gati Gayatri.pdf (accessed Apr. 10, 2023).

[5] A. O. Anindryati and I. Mufidah, Gambaran Kondisi Vitalitas Bahasa Daerah di Indonesia. 2020.

[6] Kemendikbud, “Indikator dan Cara Penanganan Kekerasan dan Bullying di Sekolah,” 2019. https://www.kemendikbud.go.id/main/blog/2019/07/indikator-dan-cara-penanganan-kekerasan-dan-bullying-di-sekolah (accessed Apr. 15, 2023).

[7] R. I. Rafiq, H. Hosseinmardi, R. Han, Q. Lv, and S. Mishra, “Scalable and timely detection of *cyberbullying* in online social networks,” Proc. ACM Symp. Appl. Comput., pp. 1738–1747, 2018, doi: 10.1145/3167132.3167317.

[8] M. Wongkar and A. Angdresey, “Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: X,” Proc. 2019 4th Int. Conf. Informatics Comput. ICIC 2019, pp. 1–5, 2019, doi: 10.1109/ICIC47613.2019.8985884.

[9] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, "A CNN-BiLSTM Model for Document-Level Sentiment Analysis," Mach. Learn. Knowl. Extr., vol. 1, no. 3, pp. 832–847, 2019, doi: 10.3390/make1030048.

[10] W. Yue and L. Li, "Sentiment Analysis using Word2vec-CNN-BiLSTM Classification," 2020 Seventh Int. Conf. Soc. Networks Anal. Manag. Secur., pp. 3–7, 2020, doi: 10.1109/SNAMS52053.2020.9336549.

[11] E. M. Dharma, F. L. Gaol, H. L. H. S. Warnars, and B. Soewito, "the Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (Cnn) Text Classification," J. Theor. Appl. Inf. Technol., vol. 100, no. 2, pp. 349–359, 2022.

[12] H. Christian, M. P. Agus, and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)," ComTech Comput. Math. Eng. Appl., vol. 7, no. 4, p. 285, 2016, doi: 10.21512/comtech.v7i4.3746.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Transactions of the Association for Computational Linguistics.," Trans. Assoc. Comput. Linguist., vol. 5, pp. 135–146, 2017, [Online]. Available: https://transacl.org/ojs/index.php/tacl/article/view/999.

[14] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, "An Efficient Text Classification Using fastText for Bahasa Indonesia Documents Classification," 2020 Int. Conf. Data Sci. Artif. Intell. Bus. Anal. DATABIA 2020 - Proc., pp. 69–75, 2020, doi: 10.1109/DATABIA50434.2020.9190447.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., pp. 1–12, 2013.

[16] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 15th Conf. Eur. Chapter Assoc. Comput. Linguist. EACL 2017 - Proc. Conf., vol. 2, pp. 427–431, 2017, doi: 10.18653/v1/e17-2068.

[17] A. Liaw and M. Wiener, "Classification and Regression by randomForest," R News, vol. 2, no. 3, pp. 18–22, 2002.

[18] K. S. Alam, S. Bhowmik, and P. R. K. Prosun, "*Cyberbullying* detection: An ensemble based machine learning approach," Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob. Networks, ICICV 2021, no. March, pp. 710–715, 2021, doi: 10.1109/ICICV50876.2021.9388499.

[19] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," Artif. Intell. Rev., vol. 52, no. 1, pp. 273–292, 2019, doi: 10.1007/s10462-018-09677-1.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[21] A. Shenfield and M. Howarth, "A novel deep learning model for the detection and identification of rolling element-bearing faults," Sensors (Switzerland), vol. 20, no. 18, pp. 1–24, 2020, doi: 10.3390/s20185112.

[22] M. Elgendy, "Human-in-the-Loop Machine Learning Version 1 MEAP Edition Manning Early Access Program Copyright 2019 Manning Publications," 2019.