


# Klasifikasi Topik Artikel Berita Menggunakan Bidirectional Long Short-Term Memory

Anyelyra Kantata<sup>1</sup>

<sup>a</sup> Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia  
<sup>1</sup> anyelyra.kantata@ui.ac.id

ARTICLE INFO		ABSTRACT
	NPM 2206048625	Dalam era digital seperti ini, jumlah artikel berita yang begitu banyak seiring berjalannya waktu, perlu diseimbangkan dengan klasifikasi topik yang lebih cepat dan akurat. Penelitian ini bertujuan untuk mengimplementasikan Bidirectional Long Short-Term Memory (BiLSTM) untuk melakukan klasifikasi topik pada artikel berita. Dataset yang diberikan terdiri dari artikel berita yang telah diberi label topik, lalu dilakukan <i>preprocessing</i> seperti tokenisasi, normalisasi, dan pembuatan <i>embedding</i> . Model kemudian dilatih dan diuji untuk mengklasifikasikan setiap artikel berita ke dalam tepat satu topik yang sesuai. Hasil evaluasi menunjukkan bahwa model memiliki akurasi sebesar 0.9147 dengan <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> yang juga sangat baik.

Kata kunci: Klasifikasi Topik, Deep Learning, Bidirectional Long Short-Term Memory

Copyright © 2025.  
All rights reserved.

## I. Pendahuluan

Pada era modern, artikel berita dipublikasikan dengan jumlah yang begitu banyak setiap hari di berbagai platform. Banyaknya data tidak terstruktur berbentuk teks menjadi tantangan bagi metode klasifikasi topik klasik yang sulit menangani data dalam jumlah besar. Klasifikasi topik adalah hal penting bagi pengambilan informasi, pembuatan rekomendasi, dan membantu untuk bisa menyusun artikel-artikel berita ke dalam topik-topik yang sudah ditentukan.

Dalam penelitian ini, penulis melakukan klasifikasi topik menggunakan Bi-LSTM yang merupakan salah satu model *deep learning* untuk mengklasifikasikan artikel-artikel berita ke dalam tepat satu topik berbeda. Bi-LSTM dapat secara langsung mempelajari pola dari input berupa teks mentah. Kontribusi utama dari penulisan penelitian ini adalah: (1) implementasi dari model Bi-LSTM yang spesifik dibuat untuk klasifikasi topik artikel berita berbahasa inggris; dan (2) evaluasi model menggunakan *overall accuracy* dan *confusion matrix*.

## II. Kajian Teori

Penelitian ini didasarkan pada beberapa riset terdahulu mengenai klasifikasi topik dengan menggunakan model *deep learning*. Beberapa riset yang menggunakan *multi-topic*, sehingga teks dapat dimasukkan ke beberapa kategori. Namun, penelitian ini hanya akan mengklasifikasikan artikel berita ke tepat satu topik. Riset yang menjadi acuan adalah klasifikasi *cyberbullying* pada *tweet* bahasa sunda dengan menggunakan beberapa model *machine learning* dan *deep learning* [1]. Riset ini bertujuan untuk mengklasifikasi *tweet* menjadi dua kelompok, yaitu *cyberbullying* dan *not cyberbullying*.

Riset ini menggunakan empat model, yaitu *Random Forest*, SVM, CNN, Bi-LSTM dan CNN-BiLSTM. Hasil pelatihan model menunjukkan bahwa Bi-LSTM dan CNN-BiLSTM memiliki *F1-score* dan *accuracy* yang paling baik. Bi-LSTM menghasilkan *F1-score* = 0.97 dan *accuracy* = 97.10%, sedangkan CNN-BiLSTM menghasilkan *F1-score* = 0.97 dan *accuracy* = 97.30%. Hasil performa dari *hybrid learning* menunjukkan bahwa CNN dapat mengekstraksi fitur dengan baik

dan BiLSTM mampu mempertahankan keterkaitan dan urutan dalam dua arah untuk memahami konteks.

### III. Data and Metode

#### A. Data.

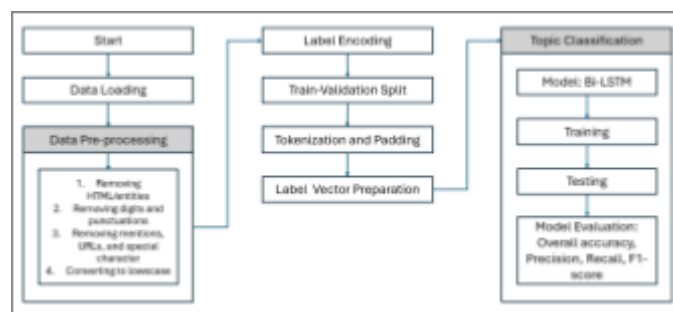
Data yang digunakan pada penelitian ini adalah dataset AG's News Topic Classification. Dataset ini menyediakan 120000 artikel untuk *training* (30000 artikel per topik) dan 7600 artikel untuk *testing* (1900 artikel per topik). Keseluruhan artikel terdistribusi secara merata pada setiap kelas: Worlds, Sports, Business, dan Sci/Tech.

Tabel 1. Deskripsi Variabel

Variable	Description
Class Index	Terdapat label kelas 1-4; dengan 1-World, 2-Sports, 3-Business, and 4-Sci Tech
Title	Judul dari artikel berita
Description	Deskripsi dari artikel berita

#### B. Metode

Prosedur untuk melakukan klasifikasi topik terdapat pada diagram di Gambar 1. Setiap proses akan dijelaskan pada bagian selanjutnya.



Gambar 1. Diagram Prosedur

#### C. Text Cleaning

Untuk mempersiapkan input teks sebelum dilatih ke dalam model, akan dilakukan *text cleaning*. Proses ini dimulai dari menghapus tag HTML. Lalu dilanjutkan dengan penghapusan angka, tanda baca, *mention*, URL, dan karakter non-alfanumerik. Selanjutnya, seluruh teks diubah menjadi huruf kecil untuk memastikan keseragaman. Tahap ini membantu mengurangi ukuran kosa kata dan meningkatkan kemampuan generalisasi model *deep learning* [2].

Tabel 2. Hasil *Text Cleaning*

Fase	Teks
Teks mentah	With the Texans nursing a second-half lead, the stage was set for another Packers' comeback, authored by Brett Favre. The result: a 16-13 Green Bay win.
Menghapus HTML/entities	With the Texans nursing a second-half lead, the stage was set for another Packers' comeback, authored by Brett Favre. The result: a 16-13 Green Bay win.
Menghapus <i>digits</i> dan <i>punctuations</i>	With the Texans nursing a secondhalf lead the stage was set for another Packers comeback authored by Brett Favre The result a Green Bay win
Menghapus <i>mentions</i> , URLs, dan <i>special characters</i>	With the Texans nursing a secondhalf lead the stage was set for another Packers comeback authored by Brett Favre The result a Green Bay win
Konversi ke huruf kecil	with the texans nursing a secondhalf lead the stage was set for another packers comeback authored by brett favre the result a green bay win

#### D. Label Encoding

Setiap label kelas pada dataset telah diberi label dari 1 hingga 4. Label ini kemudian dikonversi ke dalam format numerik dengan LabelEncoder dari scikit-learn. Tahap ini bertujuan untuk membuat label menjadi sesuai untuk model yang memerlukan input numerik dalam *training* dan evaluasi.

#### E. Train-Validation Split

Dataset dibagi menjadi subset *training* dan validasi menggunakan pembagian 80:20, hal ini bertujuan untuk mengevaluasi kinerja model dan mengurangi *overfitting*.

#### F. Tokenization and Padding

Teks yang telah melalui tahap *text cleaning*, kemudian ditokenisasi menggunakan Tokenizer dari TensorFlow Keras dengan ukuran kosakata sebesar 8192 dan token untuk kata di luar kosakata (out-of-vocabulary/OOV). Setiap kalimat diubah menjadi urutan bilangan bulat, lalu dilakukan *padding* dan *truncating* hingga panjang tetap sebanyak 128 token. *Padding* diterapkan di akhir untuk mempertahankan urutan kata. *Padding* juga memastikan bahwa semua urutan input memiliki bentuk yang sama, sehingga hal ini memungkinkan model dapat memproses input dalam *batch* sambil menjaga konsistensi terstruktur [3].

#### G. Label Vector Preparation

Label kelas untuk set *training*, validasi, dan *testing* disimpan dalam vektor terpisah ( $y_{train}$ ,  $y_{val}$ , dan  $y_{test}$ ). Tahap ini penting untuk *supervised learning* karena dibutuhkan korespondensi satu-satu antar vektor input dan label.

#### H. Bidirectional Long Short-Term Memory (Bi-LSTM)

Long Short-Term Memory (LSTM) termasuk dalam kategori arsitektur pada Recurrent Neural Networks (RNN) yang secara efektif memanfaatkan sel memori. Tujuan utama dari model LSTM adalah untuk mengatasi *gradient vanishing* dan *exploding* [putri]. Bi-LSTM adalah model yang terdiri dari dua jaringan LSTM, yang masing-masing dapat membaca input secara maju (*forward*) dan mundur (*backward*). Model ini adalah model jaringan saraf yang populer karena mampu mempertimbangkan informasi konteks dalam pembelajaran informasi secara berurutan [4].

#### I. Evaluation

Pada penelitian ini, model dievaluasi dengan mengukur *F1-score*, *recall*, *precision*, dan *accuracy*. Nilai akurasi yang digunakan pada penelitian ini dihitung dengan formula yang terdapat pada riset [5].

$$Recall = \frac{TP}{FN+TP} \quad (1)$$

$$Precision = \frac{TP}{FP+TP} \quad (2)$$

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

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

### IV. Hasil dan Pembahasan

Pada penelitian ini, diterapkan model Bi-LSTM dengan menggunakan *sparse categorical crossentropy* sebagai fungsi loss untuk mengukur seberapa jauh output prediksi model dari label

sebenarnya. Model ini menggunakan *optimizer* Adam. Pada Tabel 3, tertera *layer* pada model yang digunakan.

Tabel 3. Rangkuman Model Bi-LSTM

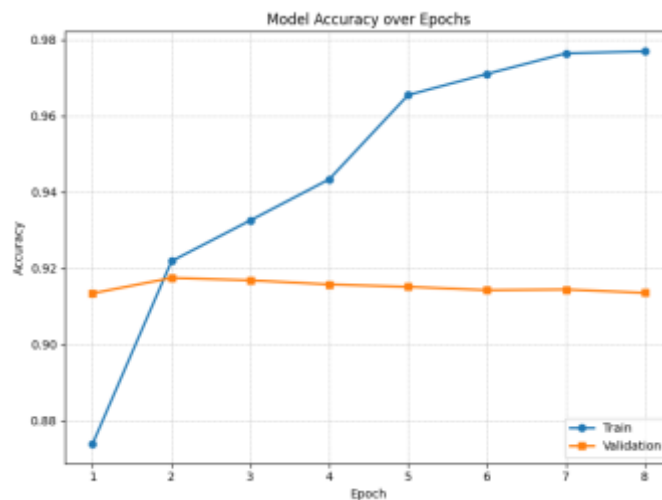
Layer (type)	Output Shape	Param #
embedding_28 (Embedding)	(None, 64, 64)	5251840
bidirectional_18 (Bidirectional)	(None, 64, 256)	197632
global_max_pooling1d_19 (GlobalMaxPooling1D)	(None, 256)	0
dense_64 (Dense)	(None, 256)	65792
dropout_60 (Dropout)	(None, 256)	0
dense_65 (Dense)	(None, 4)	1028

**Total params:** 5516292 (21.04 MB)

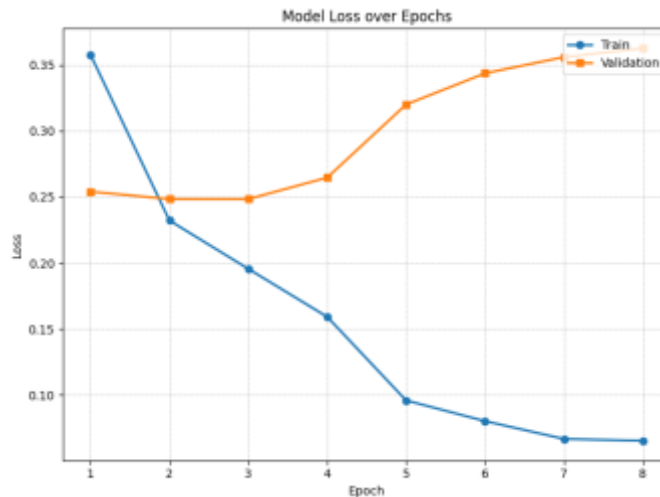
**Trainable params:** 5516292 (21.04 MB)

**Non-trainable params:** 0 (0.00 B)

Proses pelatihan ini dilakukan dengan maksimal 50 *epoch* dengan ukuran *batch* sebesar 32. Untuk mencegah *overfitting* dan mempercepat proses *training*, digunakan dua *callback* utama, yaitu EarlyStopping dan ReduceLROnPlateau. Pada Gambar 2 dan Gambar 3 terdapat *accuracy curve* dan *loss curve*. *Training accuracy* secara konsisten meningkat dari 87% hingga mencapai 98%, sementara *validation accuracy* relatif stabil. *Loss curve* menunjukkan bahwa *loss* pada data pelatihan menurun secara bertahap di setiap *epoch*.



Gambar 2. Accuracy Curve



Gambar 3. Loss Curve

Lalu, model dievaluasi pada data *testing* dan menghasilkan akurasi sebesar 91.47% dan *loss* sebesar 0.2608. *Confusion matrix* dan *classification report* menunjukkan bahwa kerja model pada setiap kelas sangat baik. Pada Tabel 4 diberikan nilai-nilai *precision*, *recall*, dan *F1-score* pada setiap kelas, hasilnya menunjukkan seluruh nilai berkisar di atas 0.87 dan hal ini menunjukkan bahwa model cukup seimbang tanpa adanya bias signifikan terhadap kelas tertentu.

Tabel 4. Classification Report

Kelas	Precision	Recall	F1-Score	Support
0	0.93	0.91	0.92	1900
1	0.96	0.98	0.97	1900
2	0.87	0.89	0.88	1900
3	0.89	0.88	0.89	1900

## V. Kesimpulan

Penelitian ini mengimplementasikan model deep learning, yaitu Bidirectional Long Short-Term Memory (Bi-LSTM), untuk melakukan klasifikasi topik pada artikel berita berbahasa Inggris menggunakan dataset AG's News Topic Classification. Model yang dikembangkan berhasil mencapai akurasi tinggi sebesar 91.47% pada data *testing*. Selain itu, nilai *precision*, *recall*, dan *F1-score* di setiap kelas berada di atas 0.87, yang menunjukkan bahwa model mampu mengklasifikasikan artikel berita ke dalam satu topik dengan cukup seimbang dan tanpa bias yang berarti. Penerapan teknik EarlyStopping dan ReduceLROnPlateau juga membantu mengendalikan potensi *overfitting* selama proses pelatihan.

## Referensi

- [1] Setyaningrum, Anisa Putri, and Muhammad Fahmy Nadhif. "Klasifikasi Cyberbullying Pada Tweet Bahasa Sunda Dengan Menggunakan Hybrid Learning Model." *Rekayasa Hijau: Jurnal Teknologi Ramah Lingkungan* 9, no. 1 (2025): 58-69.
- [2] Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released January 12, 2025. <https://web.stanford.edu/~jurafsky/slp3>.
- [3] F. Chollet, *Deep Learning with Python*, 2nd ed., Manning Publications, 2021.

- [4] Y. K. Siregar and E. B. Setiawan, "Classification of Topics Using Bi-LSTM and CNN with the Feature Expansion on Twitter," *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, Malang, Indonesia, 2023, pp. 245-250, doi: 10.1109/COMNETSAT59769.2023.10420473.
- [5] B. A. Putri and E. B. Setiawan, "Topic Classification Using the Long Short-Term Memory (LSTM) Method with FastText Feature Expansion on Twitter," *2023 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, Indonesia, 2023, pp. 18-23, doi: 10.1109/ICoDSA58501.2023.10277033.

Tautan Google Colab Notebook: [🔗 ADTT\\_B\\_2206048625\\_Anyelyra Kantata.ipynb](#)