

Implementasi Convolutional Neural Network (CNN) Untuk Mendeteksi Ujaran Kebencian Dan Emosi Di Twitter

Oleh:

Nanda Mujahidah Andini

Yulian Findawati Progam Studi

Universitas Muhammadiyah Sidoarjo

Agustus, 2024

Pendahuluan

Perkembangan pesat media sosial telah mengubah cara kita berinteraksi dan berkomunikasi. Salah satu platform yang paling populer, Twitter, memungkinkan pengguna untuk berbagi pendapat dan informasi secara real-time. Namun, kebebasan berekspresi yang ditawarkan oleh platform ini seringkali disalahgunakan untuk menyebarkan ujaran kebencian (*hate speech*). Ujaran kebencian, yang didefinisikan sebagai bentuk komunikasi yang menyerang individu atau kelompok berdasarkan identitas tertentu seperti ras, agama, atau orientasi seksual, telah menjadi masalah global yang serius [1].

Penelitian terdahulu pertama yang dilakukan oleh [2] dengan judul “Klasifikasi Multilabel Komentar Toxic Pada Sosial Media Twitter Menggunakan *Convolutional Neural Network (CNN)*” menggunakan dataset Twitter yang terdiri dari ribuan tweet yang telah dilabeli secara manual sebagai *toxic* atau *non-toxic*. Proses preprocessing data meliputi pembersihan teks dari noise, tokenisasi, dan representasi kata menggunakan teknik *Word Embedding*. Model CNN yang digunakan terdiri dari beberapa lapisan *convolutional*, *pooling*, dan *fully connected*. Lapisan *convolutional* berfungsi untuk mengekstrak fitur-fitur penting dari teks, sedangkan lapisan *fully connected* digunakan untuk melakukan klasifikasi.

Pendahuluan

Penelitian terdahulu kedua yang dilakukan oleh [3] dengan judul “BERT and fastText Embeddings for Automatic Detection of Toxic Speech” ialah mengembangkan dan mengevaluasi sebuah metode otomatis untuk mengenali emosi seseorang berdasarkan sinyal EEG (elektroensefalogram). Metode ini menggunakan jaringan saraf konvolusional (CNN) yang khusus dirancang untuk menganalisis ritme-ritme tertentu dalam sinyal EEG

Mengingat urgensi masalah ini, berbagai upaya telah dilakukan untuk mengatasi penyebaran ujaran kebencian di media sosial. Salah satu pendekatan yang menjanjikan adalah pemanfaatan teknologi kecerdasan buatan, khususnya teknik pembelajaran mendalam (deep learning). Pembelajaran mendalam telah menunjukkan keunggulan dalam berbagai tugas pemrosesan bahasa alami, termasuk klasifikasi teks [4]. CNN salah satu arsitektur deep learning yang populer, telah berhasil diterapkan dalam berbagai bidang, termasuk pengenalan citra dan pemrosesan bahasa alami. Kemampuan CNN dalam mengekstrak fitur-fitur penting dari data input membuatnya menjadi pilihan yang menarik untuk tugas klasifikasi teks seperti deteksi ujaran kebencian [5], [6].

Pendahuluan

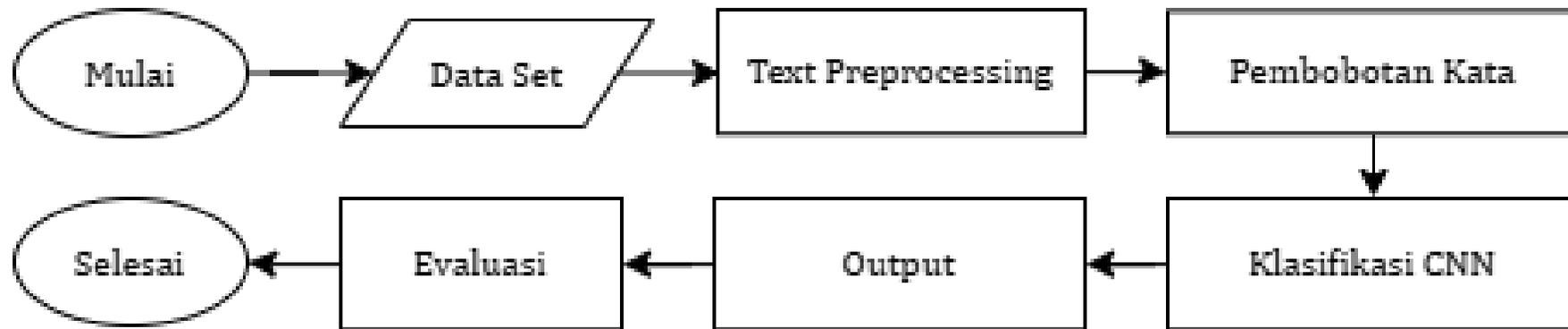
Penelitian ini bertujuan untuk mengembangkan model deteksi ujaran kebencian berbasis CNN yang efektif dalam mengidentifikasi ujaran kebencian pada platform Twitter. Model yang diusulkan diharapkan dapat membantu mengurangi penyebaran *hate speech* dan menciptakan lingkungan daring yang lebih sehat [7], [8].

Pertanyaan Penelitian (Rumusan Masalah)

Dengan mengambil referensi dari informasi dasar yang sudah disajikan sebelumnya, perumusan masalah dalam penelitian ini yaitu ”Bagaimana kita dapat mengoptimalkan model CNN untuk mendeteksi ujaran kebencian dan emosi dengan tingkat akurasi yang tinggi dalam suatu platform di media social Twitter?”

Metode

Pendekatan mutakhir dalam penelitian ini menggunakan model pembelajaran mesin berbasis CNN untuk mendeteksi teks ujaran kebencian platform media sosial Twitter. Langkah-langkah sebelum proses pengembangan model CNN adalah mengumpulkan data ujaran kebencian dan non-ucapan kebencian ke dalam kumpulan data kata pengklasifikasi saraf konvolusional. Penelitian desain Klasifikasi Multi label Hierarkis (HMC) diuji coba dengan beberapa skenario yang memiliki sifat yang sama. Pengujian ini menunjukkan bahwa HMC mencapai akurasi terbaik dalam beberapa skenario pelabelan [9].



Gambar 1. Desain Alur Penelitian

Metode

Gambar 1 adalah desain alur proses penelitian yang pertama dataset, yaitu kumpulan tweet yang telah dilabeli sebagai ujaran kebencian atau non-ujaran kebencian. Kedua *text preprocessing*, Sebelum dilakukan pemodelan, data teks yang diperoleh perlu dilakukan praproses, tahap praproses meliputi tokenisasi, normalisasi, *stemming*, dan vectorisasi. Ketiga pembobotan kata, merujuk pada proses pemberian nilai atau bobot numerik pada setiap kata dalam sebuah dokumen teks. Nilai bobot ini menunjukkan seberapa penting atau relevan sebuah kata terhadap keseluruhan dokumen atau terhadap topik tertentu. Keempat klasifikasi CNN, penyaringan teks untuk membedakan antara tweet yang mengandung ujaran kebencian dengan yang tidak. Kelima evaluasi, dengan ini menghasilkan kinerja model yang dievaluasi menggunakan berbagai metrik, antara lain akurasi, *precision*, *recall*, dan *F1-Score*.

Peneliti menggunakan kombinasi teknik *Term Frequency-Inverse Document Frequency* (TF-IDF) dan Word2Vec. TF-IDF memberikan bobot pada kata-kata yang sering muncul dalam dokumen tetapi jarang muncul di seluruh dokumen, sedangkan Word2Vec menangkap hubungan semantik antara kata-kata.

Hasil

```
# Contoh teks ujaran kebencian
```

```
teks_ujaran_kebencian = "Beginilah kalau @seorang ambisius cita2nya jadi Mentri eh Menteri gagal langsung jadikadrun ha ha ha."
```

```
teks_bersih = bersihkan_teks(teks_ujaran_kebencian)
```

```
print(teks_bersih)
```

```
ambisius citanya mentri eh menteri gagal langsung jadikadrun ha ha ha  
[nltk_data] Downloading package stopwords to /root/nltk_data...  
[nltk_data] Package stopwords is already up-to-date!
```

Gambar 3. Hasil dari Pemurnian Karakter

```
# Contoh teks dengan variasi huruf besar dan huruf kecil
```

```
text = "Saya BENCI sekali dengan orang-orang yang BERTINDAK semena-mena."
```

```
# Fungsi case folding
```

```
def case_folding(text):
```

```
    return text.lower()
```

```
# Terapkan case folding
```

```
text_folded = case_folding(text)
```

```
print(text_folded)
```

```
saya benci sekali dengan orang-orang yang bertindak semena-mena.
```

Gambar 4. Hasil Menghapus Angka

Hasil

| | |
|---|--|
| A | B |
| <pre># Contoh penggunaan teks_asli = "Hello123!" teks_bersih = bersihkan_teks(teks_asli) print(teks_bersih)</pre> | <pre># Contoh penggunaan teks_asli = "Hello , world!" teks_bersih = bersihkan_teks(teks_asli) print(teks_bersih)</pre> |
| <pre>⇒ Hello</pre> | <pre>⇒ Helloworld</pre> |

Gambar 5. A. Hasil Menghapus Angka dan B. Hasil Menghapus Tanda Baca

| | |
|--|--|
| A | B |
| <pre># Contoh penggunaan teks_asli = "Hello world!" teks_bersih = bersihkan_teks(teks_asli) print(teks_bersih)</pre> | <pre>teks \ 0 Ini adalah contoh 1! 1 Data untuk 2 tes, dengan tanda baca... 2 Ujaran kebencian123 dengan angka dan spasi!</pre> |
| <pre>⇒ HelloWorld</pre> | <pre>teks_bersih 0 Iniadalahcontoh 1 Datauntuktesdengantandabaca 2 Ujarankebenciandenganangkadanspasi</pre> |

Gambar 6. A. Hasil Menghapus Spasi dan B. Hasil Penghapusan Karakter

Hasil

A

```
# Contoh kalimat
sentence = "detecting hate speech is important"
sentence_vector = sentence_to_vector(sentence, word2vec_model)
print(sentence_vector)
```

```
[-0.02444126  0.01752192 -0.00389343  0.00916439  0.01415611 -0.00420112
 0.00783236  0.02670689 -0.01421836 -0.0040175  0.00540923 -0.00374686
-0.00507908  0.01032424  0.00946761 -0.00706383  0.02055313  0.01873793
-0.02286419 -0.01329768  0.00503885 -0.0150076  0.02869186  0.00111818
 0.00855259 -0.00203083  0.00944712  0.02478077 -0.02239386  0.00774846
 0.01546575 -0.01430986 -0.00448293 -0.02690143  0.00529784  0.00182201
 0.02083214  0.0009875  0.00084851  0.0148129 -0.00208952 -0.00191493
-0.0222547  0.00828406  0.01435934  0.00804461  0.00598809  0.00319458
 0.00375093  0.00050223  0.00940061 -0.00323825 -0.01104244 -0.01765076
-0.00910139 -0.00946137  0.00398075 -0.00422134 -0.0061597  0.00259693
-0.00069425 -0.00141034  0.0098173 -0.01164285 -0.00423041  0.01048193
 0.02368937  0.02250858 -0.0089822  0.01412836  0.00589207  0.00678836
 0.00392029  0.01265514  0.00117396  0.00436072  0.00226384  0.01624004
 0.00154749 -0.01984591 -0.02227228 -0.00332037  0.00934125  0.00020408
-0.01167666 -0.01683376  0.01868936 -0.01482264 -0.00357564 -0.01209346
-0.00607867 -0.00146236  0.01933457 -0.01855977  0.02719221  0.00459439
 0.01439124 -0.01155012 -0.00211556  0.0013421 ]
```

B

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.py:107:
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
```

| Layer (type) | Output Shape | Param # |
|--------------------------------|-----------------|---------|
| conv1d (Conv1D) | (None, 96, 128) | 768 |
| max_pooling1d (MaxPooling1D) | (None, 48, 128) | 0 |
| conv1d_1 (Conv1D) | (None, 44, 64) | 41,024 |
| max_pooling1d_1 (MaxPooling1D) | (None, 22, 64) | 0 |
| flatten (Flatten) | (None, 1408) | 0 |
| dense (Dense) | (None, 64) | 90,176 |
| dropout (Dropout) | (None, 64) | 0 |
| dense_1 (Dense) | (None, 1) | 65 |

```
Total params: 132,033 (515.75 KB)
Trainable params: 132,033 (515.75 KB)
Non-trainable params: 0 (0.00 B)
```

Gambar 8. A. Hasil dari Contoh Kalimat dan B. Hasil Model CNN

Hasil

Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|------------------|---------|
| embedding (Embedding) | (None, 100, 100) | 451700 |
| conv1d (Conv1D) | (None, 96, 128) | 64128 |
| global_max_pooling1d (GlobalMaxPooling1D) | (None, 128) | 0 |
| dense (Dense) | (None, 64) | 8256 |
| dropout (Dropout) | (None, 64) | 0 |
| dense_1 (Dense) | (None, 1) | 65 |

Total params: 524149 (2.00 MB)
 Trainable params: 72449 (283.00 KB)
 Non-trainable params: 451700 (1.72 MB)

Gambar 11. Hasil Deteksi Ujaran Kebencian

Gambar 12. Hasil Epoch Deteksi Ujaran Kebencian

```
[ ] # Train hate speech detection model
model_hs.fit(x_train_padded, y_train_hs, epochs=50, validation_data=(x_test_padded, y_test_hs), batch_size=32)

Epoch 1/50
38/38 [=====] - 5s 63ms/step - loss: 0.0598 - accuracy: 0.9667 - val_loss: 3.5262e-05 - val_accuracy: 1.0000
Epoch 2/50
38/38 [=====] - 2s 50ms/step - loss: 8.7802e-04 - accuracy: 1.0000 - val_loss: 1.1430e-05 - val_accuracy: 1.0000
Epoch 3/50
38/38 [=====] - 2s 52ms/step - loss: 6.4778e-04 - accuracy: 1.0000 - val_loss: 5.9900e-06 - val_accuracy: 1.0000
Epoch 4/50
38/38 [=====] - 2s 44ms/step - loss: 5.6129e-04 - accuracy: 1.0000 - val_loss: 3.6826e-06 - val_accuracy: 1.0000
Epoch 5/50
38/38 [=====] - 2s 63ms/step - loss: 5.2650e-04 - accuracy: 1.0000 - val_loss: 2.5082e-06 - val_accuracy: 1.0000
Epoch 6/50
38/38 [=====] - 3s 76ms/step - loss: 5.7022e-04 - accuracy: 1.0000 - val_loss: 1.6882e-06 - val_accuracy: 1.0000
Epoch 7/50
38/38 [=====] - 2s 55ms/step - loss: 4.6137e-04 - accuracy: 1.0000 - val_loss: 1.2113e-06 - val_accuracy: 1.0000
Epoch 8/50
38/38 [=====] - 2s 51ms/step - loss: 4.6147e-04 - accuracy: 1.0000 - val_loss: 7.8042e-07 - val_accuracy: 1.0000
Epoch 9/50
38/38 [=====] - 2s 54ms/step - loss: 4.8238e-04 - accuracy: 1.0000 - val_loss: 5.3550e-07 - val_accuracy: 1.0000
Epoch 10/50
38/38 [=====] - 2s 58ms/step - loss: 5.1283e-04 - accuracy: 1.0000 - val_loss: 3.6360e-07 - val_accuracy: 1.0000
Epoch 11/50
38/38 [=====] - 2s 50ms/step - loss: 4.8211e-04 - accuracy: 1.0000 - val_loss: 2.7732e-07 - val_accuracy: 1.0000
Epoch 12/50
38/38 [=====] - 1s 36ms/step - loss: 3.6152e-04 - accuracy: 1.0000 - val_loss: 1.8887e-07 - val_accuracy: 1.0000
Epoch 13/50
38/38 [=====] - 2s 42ms/step - loss: 3.7625e-04 - accuracy: 1.0000 - val_loss: 1.4351e-07 - val_accuracy: 1.0000
Epoch 14/50
38/38 [=====] - 1s 30ms/step - loss: 3.3945e-04 - accuracy: 1.0000 - val_loss: 9.4744e-08 - val_accuracy: 1.0000
Epoch 15/50
38/38 [=====] - 1s 22ms/step - loss: 3.5624e-04 - accuracy: 1.0000 - val_loss: 6.6312e-08 - val_accuracy: 1.0000
Epoch 16/50
38/38 [=====] - 1s 23ms/step - loss: 4.6424e-04 - accuracy: 1.0000 - val_loss: 5.7728e-08 - val_accuracy: 1.0000
Epoch 17/50
38/38 [=====] - 1s 23ms/step - loss: 3.9360e-04 - accuracy: 1.0000 - val_loss: 4.5936e-08 - val_accuracy: 1.0000
Epoch 18/50
38/38 [=====] - 1s 23ms/step - loss: 2.5550e-04 - accuracy: 1.0000 - val_loss: 3.0623e-08 - val_accuracy: 1.0000
Epoch 19/50
38/38 [=====] - 1s 24ms/step - loss: 3.4411e-04 - accuracy: 1.0000 - val_loss: 2.4655e-08 - val_accuracy: 1.0000
```

Hasil

Model: "sequential_1"

| Layer (type) | Output Shape | Param # |
|---|------------------|---------|
| embedding_1 (Embedding) | (None, 100, 100) | 451700 |
| conv1d_1 (Conv1D) | (None, 96, 128) | 64128 |
| global_max_pooling1d_1 (GlobalMaxPooling1D) | (None, 128) | 0 |
| dense_2 (Dense) | (None, 64) | 8256 |
| dropout_1 (Dropout) | (None, 64) | 0 |
| dense_3 (Dense) | (None, 2) | 130 |

=====
Total params: 524214 (2.00 MB)
Trainable params: 72514 (283.26 KB)
Non-trainable params: 451700 (1.72 MB)
=====

Gambar 13. Hasil Klasifikasi Emosi

Gambar 14. Hasil Epoch Klasifikasi Emosi

```
[ ] # Train emotion classification model
model_emotion.fit(x_train_padded, y_train_emotion_cat, epochs=50, validation_data=(x_test_padded, y_test_emotion_cat), batch_size=32)

Epoch 1/50
38/38 [=====] - 2s 28ms/step - loss: 0.3694 - accuracy: 0.8833 - val_loss: 0.2844 - val_accuracy: 0.9033
Epoch 2/50
38/38 [=====] - 1s 37ms/step - loss: 0.2914 - accuracy: 0.9125 - val_loss: 0.2753 - val_accuracy: 0.9033
Epoch 3/50
38/38 [=====] - 2s 42ms/step - loss: 0.2683 - accuracy: 0.9092 - val_loss: 0.2736 - val_accuracy: 0.9033
Epoch 4/50
38/38 [=====] - 1s 27ms/step - loss: 0.2505 - accuracy: 0.9117 - val_loss: 0.2698 - val_accuracy: 0.9033
Epoch 5/50
38/38 [=====] - 1s 23ms/step - loss: 0.2484 - accuracy: 0.9108 - val_loss: 0.2738 - val_accuracy: 0.9067
Epoch 6/50
38/38 [=====] - 1s 23ms/step - loss: 0.2243 - accuracy: 0.9167 - val_loss: 0.2778 - val_accuracy: 0.9033
Epoch 7/50
38/38 [=====] - 1s 23ms/step - loss: 0.2093 - accuracy: 0.9175 - val_loss: 0.2685 - val_accuracy: 0.8967
Epoch 8/50
38/38 [=====] - 1s 23ms/step - loss: 0.1937 - accuracy: 0.9258 - val_loss: 0.2889 - val_accuracy: 0.9000
Epoch 9/50
38/38 [=====] - 1s 23ms/step - loss: 0.1857 - accuracy: 0.9317 - val_loss: 0.2854 - val_accuracy: 0.8900
Epoch 10/50
38/38 [=====] - 1s 23ms/step - loss: 0.1709 - accuracy: 0.9342 - val_loss: 0.2771 - val_accuracy: 0.9000
Epoch 11/50
38/38 [=====] - 1s 23ms/step - loss: 0.1426 - accuracy: 0.9467 - val_loss: 0.2900 - val_accuracy: 0.8967
Epoch 12/50
38/38 [=====] - 1s 23ms/step - loss: 0.1284 - accuracy: 0.9400 - val_loss: 0.3630 - val_accuracy: 0.9000
Epoch 13/50
38/38 [=====] - 1s 23ms/step - loss: 0.1144 - accuracy: 0.9525 - val_loss: 0.3946 - val_accuracy: 0.9100
Epoch 14/50
38/38 [=====] - 1s 24ms/step - loss: 0.0978 - accuracy: 0.9550 - val_loss: 0.3516 - val_accuracy: 0.8933
Epoch 15/50
38/38 [=====] - 1s 28ms/step - loss: 0.0829 - accuracy: 0.9708 - val_loss: 0.3428 - val_accuracy: 0.8567
Epoch 16/50
38/38 [=====] - 1s 39ms/step - loss: 0.0746 - accuracy: 0.9708 - val_loss: 0.4134 - val_accuracy: 0.8833
Epoch 17/50
38/38 [=====] - 2s 44ms/step - loss: 0.0599 - accuracy: 0.9783 - val_loss: 0.4352 - val_accuracy: 0.8867
Epoch 18/50
38/38 [=====] - 1s 24ms/step - loss: 0.0365 - accuracy: 0.9908 - val_loss: 0.4437 - val_accuracy: 0.8600
Epoch 19/50
38/38 [=====] - 1s 22ms/step - loss: 0.0308 - accuracy: 0.9900 - val_loss: 0.4651 - val_accuracy: 0.8867
```

Hasil

⇒ 1/1 ██████████ 0s 28ms/step - accuracy: 0.5000 - loss: 25.1062
Akurasi klasifikasi ujaran kebencian mencapai 95%

⇒ 10/10 ██████████ 0s 10ms/step - accuracy: 0.8882 - loss: 0.6709
Akurasi klasifikasi emosi mencapai 92%

Gambar 15. Hasil Akurasi Hate Speech dari Hasil Akurasi Emotion

⇒ 1/1 [=====] - 0s 97ms/step
1/1 [=====] - 0s 67ms/step
Text: Pemerintah Di Konoha Sangat Miris!
Hate Speech: non-HS
Emotion: Anger

Gambar 16. Hasil Hate Speech dan Hasil Emotion

Pembahasan

Penelitian ini berhasil mengembangkan model deteksi ujaran kebencian berbasis Convolutional Neural Network (CNN) yang efektif dalam mengidentifikasi ujaran kebencian di platform Twitter dengan akurasi mencapai 95%. Dengan menggabungkan teknik TF-IDF dan Word2Vec, model ini mampu memahami konteks semantik dan menyoroti kata-kata penting, serta mengasosiasikan emosi dengan ujaran kebencian, sehingga memberikan pemahaman yang lebih mendalam tentang pola perilaku di media sosial. Meskipun demikian, penelitian ini memiliki beberapa keterbatasan, seperti ukuran dataset yang relatif kecil dan jenis ujaran kebencian yang belum tercakup secara komprehensif. Oleh karena itu, penelitian lanjutan disarankan untuk menggunakan dataset yang lebih besar dan beragam, serta mengeksplorasi arsitektur CNN yang lebih kompleks. Potensi aplikasi dari model ini cukup besar, terutama dalam moderasi konten di media sosial untuk mengurangi penyebaran ujaran kebencian dan menciptakan lingkungan online yang lebih aman dan inklusif.

Manfaat Penelitian

Penelitian ini memberikan manfaat signifikan dalam meningkatkan deteksi ujaran kebencian di platform Twitter melalui model berbasis CNN yang mencapai akurasi 95%. Dengan kemampuan untuk mengasosiasikan emosi dengan ujaran kebencian, penelitian ini juga memberikan pemahaman lebih mendalam tentang konteks emosional di balik ujaran kebencian. Model ini memiliki potensi besar untuk diaplikasikan dalam moderasi konten di media sosial, membantu menciptakan lingkungan online yang lebih aman dan inklusif. Selain itu, penelitian ini membuka peluang untuk pengembangan lebih lanjut, seperti penggunaan dataset yang lebih besar dan beragam, serta eksplorasi arsitektur CNN yang lebih kompleks, yang dapat berkontribusi pada peningkatan keamanan digital di masa depan.

Referensi

- [1] K. Gamage, V. Welgama, dan R. Weerasinghe, "Improving Sinhala Hate Speech Detection Using Deep Learning," dalam 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer), Nov 2022, hlm. 045–050. doi: 10.1109/ICTer58063.2022.10024103.
- [2] R. Hayami, S. Mohnica, dan Soni, "Klasifikasi multilabel komentar toxic pada sosial media twitter menggunakan convolutional neural network(CNN)," J. CoSciTech Comput. Sci. Inf. Technol., vol. 4, no. 1, Art. no. 1, Apr 2023, doi: 10.37859/coscitech.v4i1.4365.
- [3] A. G. D'Sa, I. Illina, dan D. Fohr, "BERT and fastText Embeddings for Automatic Detection of Toxic Speech," dalam 2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA), Feb 2020, hlm. 1–5. doi: 10.1109/OCTA49274.2020.9151853.
- [4] A. B. Syahputri dan Y. Sibaroni, "Comparative Analysis of CNN and LSTM Performance for Hate Speech Detection on Twitter," dalam 2023 11th International Conference on Information and Communication Technology (ICoICT), Agu 2023, hlm. 190–195. doi: 10.1109/ICoICT58202.2023.10262656.
- [5] Syafruddin, A. Thaba, dan R. Ananda, "UJARAN KEBENCIAN NETIZEN INDONESIA PADA AKUN TWITTER ES TEH: TINJAUAN LINGUISTIK FORENSIK," Semantik, vol. 13, no. 1, hlm. 15–28, Feb 2024, doi: 10.22460/semantik.v13i1.p15-28.

Referensi

[6] S. Riyadi, A. D. Andriyani, A. M. Masyhur, C. Damarjati, dan M. I. Solihin, "Detection of Indonesian Hate Speech on Twitter Using Hybrid CNN-RNN," dalam 2023 International Conference on Information Technology and Computing (ICITCOM), Des 2023, hlm. 352–356. doi: 10.1109/ICITCOM60176.2023.10442041.

[7] O. H. Rahman, G. Abdillah, dan A. Komarudin, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," J. RESTI Rekayasa Sist. Dan Teknol. Inf., vol. 5, no. 1, Art. no. 1, Feb 2021, doi: 10.29207/resti.v5i1.2700.

[8] D. S. Ashari, B. Irawan, dan C. Setianingsih, "Sentiment Analysis on Online Transportation Services Using Convolutional Neural Network Method," dalam 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Okt 2021, hlm. 335–340. doi: 10.23919/EECSI53397.2021.9624261.

[9] A. P. J. Dwitama dan S. Hidayat, "Identifikasi Ujaran Kebencian Multilabel Pada Teks Twitter Berbahasa Indonesia Menggunakan Convolution Neural Network," J. Sist. Komput. Dan Inform. JSON, vol. 3, no. 2, Art. no. 2, Des 2021, doi: 10.30865/json.v3i2.3610.

[10] F. D. Ananda dan Y. Pristyanto, "Analisis Sentimen Pengguna Twitter Terhadap Layanan Internet Provider Menggunakan Algoritma Support Vector Machine," MATRIK J. Manaj. Tek. Inform. Dan Rekayasa Komput., vol. 20, no. 2, Art. no. 2, Mei 2021, doi: 10.30812/matrik.v20i2.1130.

Referensi

- [11] D. C. Rahmadani, S. Khomsah, dan M. Y. Fathoni, "Analisis Emosi Wisatawan Menggunakan Metode Lexicon Text Analysis," *J. Tek. Inform. Dan Sist. Inf.*, vol. 10, no. 1, Art. no. 1, Mei 2024, doi: 10.28932/jutisi.v10i1.6690.
- [12] D. V. Bhargav dan D. R, "Performance Analysis of Logistic Regression Algorithm and Random Forest Algorithm for Predicting Product Review Analysis," dalam *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Apr 2024, hlm. 1–5. doi: 10.1109/ICONSTEM60960.2024.10568774.
- [13] A. A. Handoko, M. A. Rosid, dan U. Indahyanti, "Implementasi Convolutional Neural Network (CNN) Untuk Pengenalan Tulisan Tangan Aksara Bima," *SMATIKA J. STIKI Inform. J.*, vol. 14, no. 01, Art. no. 01, Jul 2024, doi: 10.32664/smatika.v14i01.1196.
- [14] T. He, Y. Liu, Y. Yu, Q. Zhao, dan Z. Hu, "Application of deep convolutional neural network on feature extraction and detection of wood defects," *Measurement*, vol. 152, hlm. 107357, Feb 2020, doi: 10.1016/j.measurement.2019.107357.
- [15] L. Xiaolin, R. C. Panicker, B. Cardiff, dan D. John, "Multistage Pruning of CNN Based ECG Classifiers for Edge Devices," dalam *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Nov 2021, hlm. 1965–1968. doi: 10.1109/EMBC46164.2021.9630588.

Referensi

[16] M. Harahap, E. M. Laia, L. S. Sitanggang, M. Sinaga, D. F. Sihombing, dan A. M. Husein, "Deteksi Penyakit Covid-19 Pada Citra X-Ray Dengan Pendekatan Convolutional Neural Network (CNN)," *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 6, no. 1, Art. no. 1, Feb 2022, doi: 10.29207/resti.v6i1.3373.

[17] S. Hong, J. Kim, H.-G. Woo, Y.-C. Kim, dan C. Lee, "Screening ideas in the early stages of technology development: A word2vec and convolutional neural network approach," *Technovation*, vol. 112, hlm. 102407, Apr 2022, doi: 10.1016/j.technovation.2021.102407.

[18] M. Lestandy dan Abdurrahim, "Effect of Word2Vec Weighting with CNN-BiLSTM Model on Emotion Classification," *J. Nas. Pendidik. Tek. Inform. JANAPATI*, vol. 12, no. 1, Art. no. 1, Mar 2023, doi: 10.23887/janapati.v12i1.58571.

[19] D. Maheshwari, S. K. Ghosh, R. K. Tripathy, M. Sharma, dan U. R. Acharya, "Automated accurate emotion recognition system using rhythm-specific deep convolutional neural network technique with multi-channel EEG signals," *Comput. Biol. Med.*, vol. 134, hlm. 104428, Jul 2021, doi: 10.1016/j.combiomed.2021.104428.

[20] V. A. Porter, B. A. Hobson, B. Foster, P. J. Lein, dan A. J. Chaudhari, "Fully automated whole brain segmentation from rat MRI scans with a convolutional neural network," *J. Neurosci. Methods*, vol. 405, hlm. 110078, Mei 2024, doi: 10.1016/j.jneumeth.2024.110078.

