Progresif: Jurnal Ilmiah Komputer
https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru Loktabat – Banjarbaru,
e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0877

Klasifikasi Teks Depresi Pada Media Sosial Menggunakan *Convolutional Neural Network* Dengan *Fasttext Embedding*

DOI: http://dx.doi.org/10.35889/progresif.v22i3.3899

Azki Muntazul Hamdi[1*], Mohammad Zoqi Sarwani[2], Anang Aris Widodo[3]
Informatika, Universitas Merdeka Pasuruan, Kota Pasuruan, Indonesia
*e-mail *Corresponding Author: muntazulhamdi@student.unmerpas.ac.id*

Abstract

Depression is a persistent mood disorder that significantly impacts an individual's functioning, and its prevalence is becoming increasingly concerning among adolescents (the "Strawberry Generation"). Expressions of this psychological distress have largely migrated to social media in the form of unstructured text posts laden with informal terms, abbreviations, and typos. These linguistic characteristics pose "Out-of-Vocabulary" (OOV) challenges for traditional classification models. This study proposes integrating FastText word embeddings—which utilize subword information—to enhance feature representation, combined with a Convolutional Neural Network (CNN) architecture as the primary classifier. Using the "Student Depression Text" secondary dataset from Kaggle (comprising 7,489 entries), class imbalance was addressed exclusively within the training data using the Random Oversampling technique. Experimental results demonstrate that the proposed model achieved an overall accuracy of 95%, with Precision, Recall, and F1-Score values for the depression class of 90%, 80%, and 84%, respectively. The integration of CNN and pre-trained FastText proved to be a reliable and efficient solution for managing the complexities of digital language in the context of early mental health detection.

Keywords: *Depression; Social Media; Natural Language Processing; Convolutional Neural Network; FastText Embedding.*

Abstrak

Depresi merupakan gangguan suasana hati persisten yang secara signifikan memengaruhi fungsi fungsional individu dan prevalensinya kian mengkhawatirkan pada kelompok remaja (Strawberry Generation). Ekspresi tekanan psikologis ini kini banyak bermigrasi ke media sosial dalam bentuk unggahan teks tidak terstruktur yang sarat akan istilah non-formal, singkatan, dan salah ketik (typo). Karakteristik bahasa tersebut memicu kendala Out-of-Vocabulary (OOV) pada model klasifikasi tradisional. Penelitian ini mengusulkan integrasi word embedding FastText berbasis informasi tingkat sub-kata (subword information) untuk memperkuat representasi fitur, yang dikombinasikan dengan arsitektur Convolutional Neural Network (CNN) sebagai pengklasifikasi utama. Menggunakan dataset sekunder "Student Depression Text" sebanyak 7.489 entri dari Kaggle, ketidakseimbangan kelas diatasi secara eksklusif pada data latih menggunakan teknik Random Oversampling. Hasil eksperimen menunjukkan bahwa model yang diusulkan meraih akurasi keseluruhan sebesar 95%, dengan nilai Precision, Recall, dan F1-Score untuk kelas depresi masing-masing sebesar 90%, 80%, dan 84%. Integrasi CNN dan pre-trained FastText terbukti efektif menjadi solusi jalan tengah yang andal dan efisien dalam menangani kompleksitas bahasa digital untuk deteksi dini kesehatan mental.

Kata kunci: *Depresi; Media Sosial; Natural Language Processing; Convolutional Neural Network; FastText Embedding.*

1. Pendahuluan

Kesehatan mental, khususnya gangguan depresi mayor (Major Depressive Disorder), telah bertransformasi menjadi tantangan kesehatan global sekaligus kontributor utama beban

disabilitas di seluruh dunia. Di Indonesia, urgensi penanganan kondisi ini kian nyata pada kelompok Strawberry Generation (remaja dan dewasa muda usia 15–24 tahun) yang tumbuh berdampingan dengan media sosial dan dinilai memiliki kerentanan emosional yang tinggi [1]. Data empiris menunjukkan situasi yang memprihatinkan, di mana lebih dari 31 juta masyarakat Indonesia berusia di atas 15 tahun mengalami gangguan emosional dan 12 juta lainnya secara klinis menderita depresi. Dinamika digital mendorong kelompok berisiko ini memanfaatkan platform media sosial sebagai ruang aman untuk mengekspresikan keluh kesah serta tekanan psikologis mereka secara terbuka dan real-time. Meskipun fenomena ini membuka peluang untuk pemantauan kondisi psikologis, proses identifikasi manual terhadap jutaan unggahan teks digital oleh tenaga medis merupakan hal yang mustahil untuk dilakukan [2].

Meskipun pemantauan media sosial sangat potensial, identifikasi kondisi depresi melalui unggahan teks menghadapi tantangan fundamental pada domain *Natural Language Processing* (NLP). Teks media sosial memiliki karakteristik yang sangat tidak terstruktur, dipenuhi kata gaul (*slang*), singkatan tidak baku, metafora emosional yang implisit, serta kesalahan penulisan (*typo*). Karakteristik bahasa non-formal ini secara langsung memicu timbulnya masalah *Out-of-Vocabulary* (OOV) pada metode ekstraksi fitur spasial dan kosa kata konvensional. Di sisi lain, proses identifikasi secara manual terhadap jutaan unggahan digital oleh tenaga medis profesional merupakan hal yang mustahil dilakukan secara *real-time* karena keterbatasan sumber daya manusia [3]. Oleh karena itu, dibutuhkan sebuah sistem klasifikasi teks otomatis berbasis kecerdasan buatan yang tidak hanya sanggup mengenali konteks dan nuansa emosional, tetapi juga kebal (*robust*) terhadap variasi bahasa tidak terstruktur [4].

Berbagai penelitian terdahulu telah berupaya mengembangkan model klasifikasi teks untuk identifikasi kesehatan mental. Rahayu dkk. [3] mengimplementasikan algoritma *Random Forest* dengan ekstraksi TF-IDF dan meraih akurasi 96%, namun metode ini mengabaikan urutan kata sehingga gagal menangkap konteks kalimat secara menyeluruh. Ridha dkk. [4] menerapkan *Support Vector Machine* (SVM) berbasis TF-IDF dan *Chi-Square* untuk mengklasifikasikan teks depresi, tetapi hanya mencatatkan akurasi 74,93% akibat keterbatasan model statistik dalam menangani bahasa implisit dan kata OOV. Untuk menangkap ketergantungan sekuensial yang lebih kompleks, Ivan Dwi Nugraha & Azhar [5] menerapkan *Long Short-Term Memory* (LSTM-RNN) dan memperoleh F1-score 86%, sementara Rahmadika dkk. [6] mengombinasikan model *Transformer* (BERT) dengan XGBoost yang menghasilkan akurasi 93,05%. Meskipun arsitektur berbasis *Transformer* dan RNN mencatatkan akurasi yang memuaskan, model tersebut membutuhkan daya komputasi yang sangat masif, waktu pelatihan yang lama, serta kerap mengalami *bottleneck* ketika dihadapkan pada kata-kata baru atau *typo* yang tidak ada dalam kamus *embedding* tradisional seperti Word2Vec atau GloVe. Kesenjangan (*gap*) utama yang masih tersisa adalah ketersediaan model klasifikasi yang tidak hanya akurat dan efisien secara komputasi, tetapi juga adaptif dalam memetakan kata-kata OOV dan *typo* yang mendominasi ekspresi depresi di media sosial.

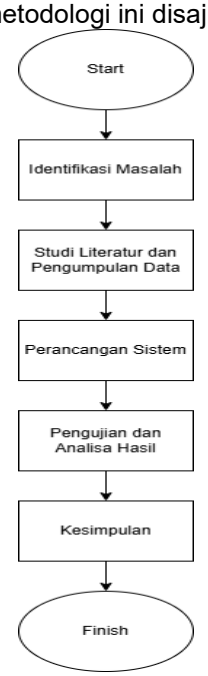
Untuk menjembatani kesenjangan (*gap*) antara efisiensi komputasi dan performa model, penelitian ini mengusulkan sebuah pendekatan terintegrasi menggunakan *Convolutional Neural Network* (CNN) dengan ekspansi fitur dari *word embedding* FastText (cc.en.300.bin). Landasan teoretis penggunaan FastText terletak pada kemampuannya mengekstrak informasi tingkat sub-kata (*subword information*) berbasis n-gram karakter [7]. Sementara itu, pemilihan arsitektur CNN didasarkan pada Teori Ekstraksi Fitur Spasial Teks yang membuktikan keunggulannya dalam mengekstraksi fitur lokal (*local patterns*) dan kombinasi frasa bermuatan emosional melalui operasi filter konvolusi 1D secara responsif dan efisien [8]. Efektivitas integrasi ini diperkuat oleh temuan empiris dari berbagai riset terkait, seperti [9], [10], serta [11], yang membuktikan bahwa penggabungan representasi sub-kata FastText dan ekstraksi spasial CNN mampu menghasilkan akurasi klasifikasi teks yang tinggi pada data tidak terstruktur dengan konsumsi daya komputasi dan memori yang jauh lebih efisien dibandingkan arsitektur *Transformer*. Oleh karena itu, konsep hibrida CNN dan FastText dipandang sebagai solusi yang sangat rasional dan efektif untuk mengatasi masalah *Out-of-Vocabulary* (OOV) sekaligus menjaga efisiensi proses deteksi depresi secara *real-time*. Melalui mekanisme morfologis ini, FastText secara inheren mampu membentuk representasi vektor numerik yang presisi untuk kata baru, kata gaul, maupun kata yang mengalami *typo* berdasarkan kemiripan sub-karakternya [7]. Melalui pendekatan morfologis ini, FastText secara inheren mampu mengatasi kendala OOV dengan tetap merekonstruksi representasi vektor bagi kata baru atau kata yang salah ketik. Tujuan utama dari penelitian ini adalah mengimplementasikan gabungan metode

CNN dan pre-trained FastText cc.en.300.bin pada dataset "Student Depression Text", guna membangun model klasifikasi biner ("Normal" vs "Kecemasan/Depresi") yang tidak hanya akurat tetapi juga efisien untuk mendukung sistem peringatan dini kesehatan mental di platform digital [6].

2. Metodologi

2.1 Alur Penelitian

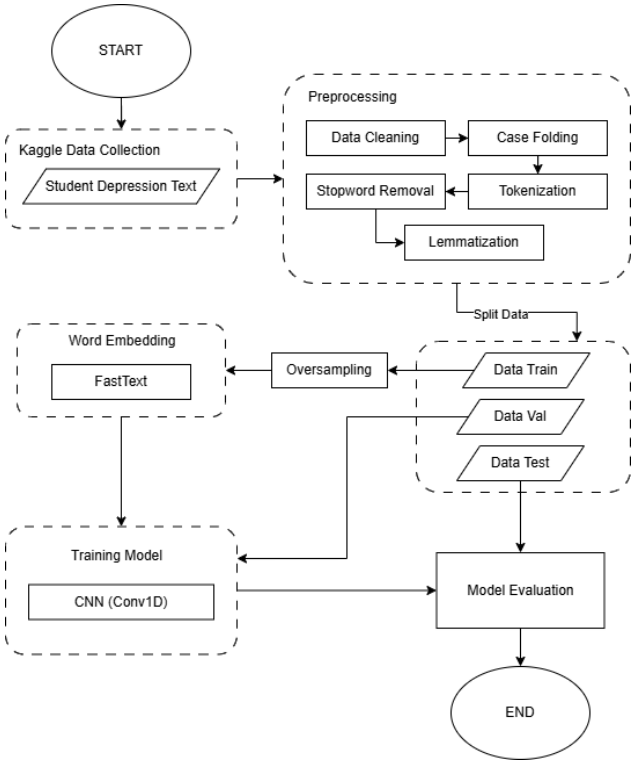
Prosedur pelaksanaan penelitian ini dilakukan secara terstruktur melalui enam tahapan utama guna menjamin reliabilitas hasil klasifikasi. Tahapan-tahapan tersebut meliputi: (1) Identifikasi Masalah, (2) Studi Literatur dan Pengumpulan Data, (3) Menentukan Metode Penelitian, (4) Perancangan Sistem, (5) Pengujian dan Analisa Hasil, serta (6) Penarikan Kesimpulan. Secara visual, runtunan metodologi ini disajikan pada diagram alur Gambar 1.



Gambar 1. Alur Penelitian

2.2 Perancangan Sistem

Arsitektur teknis pengolahan data dan pembangunan model deep learning digambarkan menggunakan flowchart terintegrasi pada **Gambar** 2. Alur ini merepresentasikan siklus data dari dokumen mentah hingga menghasilkan keputusan metrik evaluasi model.



Gambar 2. Flowchart Prosedur Penelitian

Rincian dan penjelasan detail dari setiap tahapan alur penelitian pada Gambar 1 dijelaskan secara rinci sebagai berikut.

2.3 Kaggle Data Collection

Penelitian ini menggunakan dataset sekunder "Student Depression Text" yang diperoleh dari Kaggle dengan volume total 7.489 entri data biner. Proses anotasi teks dilakukan oleh kelompok mahasiswa rentang usia 13–17 tahun yang memiliki kompetensi bahasa Inggris tingkat lanjut untuk menangkap esensi ekspresi remaja secara akurat. Struktur data terdiri dari lima kolom, yaitu Text, Label, Age, Gender, dan Age Category.

2.4 Text Preprocessing

Proses *preprocessing* teks bertujuan untuk mengubah data teks mentah yang tidak terstruktur (*unstructured raw text*) dari media sosial menjadi bentuk yang bersih, konsisten, dan siap diproses oleh algoritma *deep learning*. Pada penelitian ini, seluruh tahapan pembersihan dilakukan melalui dua fungsi utama, yaitu clean_text() dan preprocess_text(), dengan memanfaatkan pustaka re (*Regular Expression*) serta nltk (*Natural Language Toolkit*).

1. Data Cleaning dan Case Folding
   Tahap awal pemrosesan dilakukan melalui fungsi clean_text() yang bertugas mengeliminasi elemen-elemen *noise* dan menyeragamkan format teks. Teks mentah mula-mula diubah menjadi huruf kecil (text.lower()) untuk menghilangkan perbedaan antara huruf kapital dan huruf kecil. Selanjutnya, teks dibersihkan dari komponen yang tidak relevan dengan analisis emosi, seperti tautan situs web (http, https, www), sebutan akun (@user), tanda pagar (#hashtag), emotikon, tanda baca, simbol khusus, spasi berlebih, serta karakter angka.
2. Tokenization
   Teks yang telah dibersihkan kemudian diproses oleh fungsi preprocess_text(). Pada tahap ini, untaian kalimat dipotong menjadi sekumpulan entitas kata tunggal (*token*) menggunakan fungsi word_tokenize() dari pustaka NLTK. Pemotongan kata berbasis spasi ini menjadi dasar bagi penyaringan dan manipulasi kata pada tahap berikutnya.
3. Stopword Removal dan Length Filtering

Token-token kata hasil tokenisasi selanjutnya melewati proses penyaringan ganda. Penyaringan pertama dilakukan dengan menghapus kata-kata umum dalam bahasa Inggris (*stopwords*) berdasarkan kamus bawaan NLTK. Penyaringan kedua dilakukan dengan membuang kata-kata pendek yang memiliki panjang karakter kurang dari atau sama dengan dua karakter (len(word) > 2) untuk memastikan hanya kata-kata yang bermakna emosional signifikan yang dipertahankan.

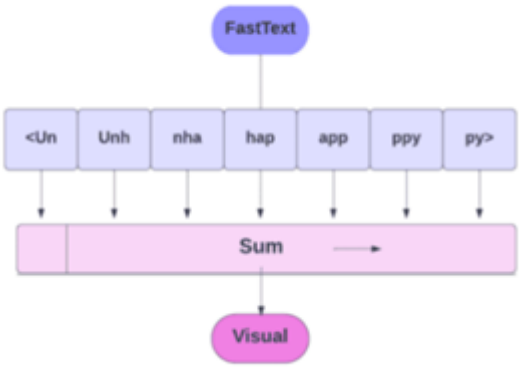
4. Lemmatization
   Token kata yang lolos seleksi diteruskan ke tahap lematisasi menggunakan fungsi WordNetLemmatizer() dari NLTK. Tahap ini bertugas mengembalikan kata-kata berimbuhan kembali ke bentuk kata dasarnya (*lemma*) berdasarkan tata bahasa Inggris, seperti mengubah kata *"depressed"* atau *"depressing"* menjadi kata dasar *"depress"*.
5. Reconstruction dan Output Filtering
   Tahap akhir dari *preprocessing* dilakukan dengan merekonstruksi kembali token-token lema yang telah bersih menjadi satu kesatuan string kalimat utuh menggunakan pemisah spasi (' '.join(tokens)). Setelah seluruh teks diproses, baris data yang menghasilkan string kosong dieliminasi dari dataset untuk menjaga validitas dan kestabilan pelatihan model.

2.5 Pembagian Data dan Penanganan Imbalance Data

Dataset yang telah bersih dibagi (data splitting) menjadi tiga bagian independen menggunakan rasio 80% data latih (training data), 10% data validasi (validation data), dan 10% data uji (testing data). Guna mengatasi bias kelas mayoritas, teknik Random Oversampling via pustaka imblearn (RandomOverSampler) diterapkan secara eksklusif hanya pada data latih. Strategi isolasi ini krusial untuk mencegah terjadinya kebocoran data (data leakage), di mana informasi dari data uji atau validasi tidak boleh mengontaminasi proses pelatihan model.

2.6 Ekstraksi Fitur FastText dan Arsitektur CNN

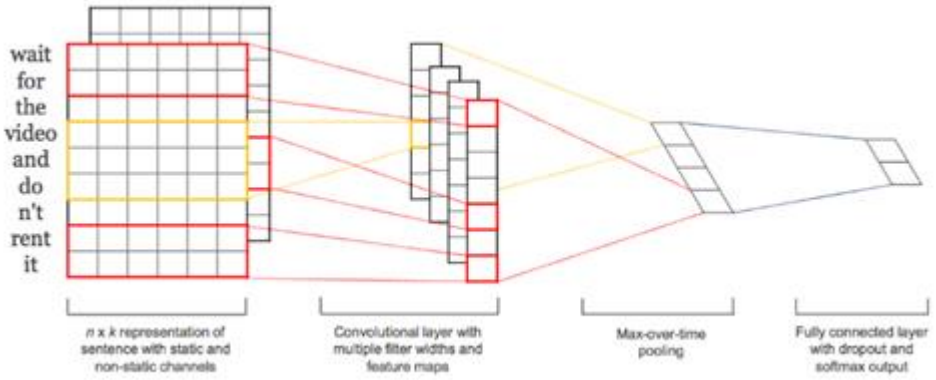
Pendekatan klasifikasi teks pada penelitian ini menggabungkan dua mekanisme utama, yaitu pembentukan representasi kata berbasis *subword information* menggunakan FastText serta ekstraksi fitur spasial lokal menggunakan Convolutional Neural Network (CNN). Integrasi ini dirancang khusus untuk mengatasi masalah kata di luar kamus (*Out-of-Vocabulary*/OOV) dan *typo* sebelum dialirkan ke dalam jaringan konvolusi 1D [12]. Mekanisme pemetaan kata menjadi vektor sub-kata menggunakan FastText diilustrasikan pada **Gambar** 3.



Gambar 3. Arsitektur Convolutional Neural Network untuk Klasifikasi Teks

FastText tidak memperlakukan kata sebagai entitas tunggal yang utuh, melainkan memecahnya menjadi potongan *character n-grams* (misalnya n-gram berukuran 3 untuk kata *"Unhappy"* dipecah menjadi <Un, Unh, app, hap, nha, ppy, py>). *Convolutional Neural Network* (CNN) merupakan salah satu arsitektur *deep learning* yang sangat efektif dalam mengekstrak fitur lokal dan pola spasial dari data sequential seperti teks. Dalam pemrosesan bahasa alami (*Natural Language Processing*), CNN memanfaatkan operasi konvolusi satu dimensi (1D) untuk menggeser filter (*kernel*) di sepanjang sekuens kata guna menangkap kombinasi frasa (*n-gram*)

bermuatan makna emosional. Gambaran visual mengenai alur dan komponen utama dari arsitektur CNN untuk klasifikasi teks disajikan pada Gambar 4.



Gambar 4. Arsitektur Convolutional Neural Network untuk Klasifikasi Teks [8]

Token teks dari data latih dikonversi menjadi vektor numerik menggunakan model pre-trained FastText cc.en.300.bin dengan dimensi vektor kontinu sebesar 300. Lapisan embedding ini dikonfigurasi dalam status terkunci (frozen/trainable=False) untuk menjaga integritas bobot semantik orisinal FastText.

Vektor tersebut kemudian dialirkan ke dalam arsitektur *Convolutional Neural Network* (CNN) dengan spesifikasi teknis penyeimbang performa dan efisiensi sebagai berikut:

1. SpatialDropout1D (0.3): Diterapkan tepat setelah layer embedding untuk mematikan seluruh saluran fitur secara acak guna meminimalisir risiko overfitting.
2. Convolutional Layer (Conv1D): Menggunakan 64 filter dengan ukuran kernel sebesar 3 (trigram) dan fungsi aktivasi ReLU. Lapisan ini mengekstraksi hubungan spasial lokal dari kombinasi setiap 3 kata berurutan yang membawa muatan psikologis, diperkuat oleh regularisasi L2 Regularizer (0.001).
3. Global MaxPooling 1D: Mengambil nilai fitur paling dominan dan kuat dari setiap peta filter konvolusi untuk merampingkan dimensi data.
4. Fully Connected & Output Layer: Mengalirkan fitur ke lapisan Dense (32 unit, ReLU) dengan dropout 0,5, dan diakhiri oleh output layer dengan 1 unit saraf berbasis fungsi aktivasi Sigmoid untuk menghasilkan probabilitas klasifikasi biner.

Proses komputasi model dioptimalkan menggunakan optimizer Adam dan loss function Binary Crossentropy. Stabilitas iterasi dikontrol secara adaptif menggunakan callback Early Stopping (patience=3) dan ReduceLROnPlateau dengan total batas atas pelatihan sebanyak 30 epoch dan ukuran batch 64.

3. Hasil dan Pembahasan

Bab ini menyajikan implementasi, analisis hasil, dan pembahasan eksperimen klasifikasi teks depresi berbasis integrasi FastText dan *Convolutional Neural Network* (CNN). Penyajian hasil disusun secara sistematis mengikuti lima tahapan utama pada alur penelitian, yaitu pengumpulan data, *text preprocessing* dan pembagian data, ekstraksi fitur FastText dan pemodelan CNN, evaluasi dan analisis kesalahan (*error analysis*), serta penarikan kesimpulan operasional.

3.1 Karakteristik Dataset

Proses pengumpulan data menggunakan dataset sekunder bertajuk *"Student Depression Text"* yang diperoleh dari repositori Kaggle (berasal dari media sosial berbasis teks seperti Twitter/Reddit). Rincian karakteristik dataset yang digunakan dirangkum pada Tabel 1.

Tabel 1. Karakteristik Dataset Penelitian

Karakteristik	Keterangan
Sumber Data	Media Sosial (Twitter/Reddit) via Repositori Kaggle
Bahasa Utama	Bahasa Inggris (*English*)
Jumlah Data Awal	7.486 baris teks mentah
Jumlah Data Setelah Pembersihan	7.452 baris teks valid
Distribusi Kelas Awal	• Label 0 (Non-Depresi / Normal): 6.259 sampel (83,6%) • Label 1 (Depresi / Anxiety): 1.227 sampel (16,4%)
Karakteristik Teks Mentah	Mengandung bahasa gaul (*slang*), kata *typo*, tanda baca berlebih, URL, *mention*, serta *Out-of-Vocabulary* (OOV).

3.2 Konfigurasi Arsitektur Model CNN yang Digunakan

Eksperimen pelatihan model *deep learning* dalam penelitian ini dibangun menggunakan bahasa pemrograman Python dengan pustaka utama TensorFlow/Keras untuk penyusunan arsitektur *Convolutional Neural Network* (CNN). Proses konversi teks menjadi representasi vektor numerik kontinu memanfaatkan model *pre-trained* FastText cc.en.300.bin dengan dimensi vektor sebesar 300.

Guna menjaga konsistensi dan transparansi eksperimen, seluruh pengaturan nilai parameter (*hyperparameter*) yang ditanamkan di dalam kode program dirangkum ke dalam Tabel 2 berikut:

Tabel 2. Parameter dan Konfigurasi Model CNN-FastText

Parameter	Konfigurasi
Embedding	FastText cc.en.300.bin (300 Dimensi)
Trainable State	False
Pooling Type	Global Max Pooling 1D
Activation Function	Sigmoid
Optimizer	Adam
Batch Size	64
Maximum Epochs	30
Callbacks	Early Stopping, Reduce LR on Plateau

3.3 Hasil Fase 1: Identifikasi Masalah dan Pengumpulan Data

Proses eksperimen diawali dengan pengumpulan dataset sekunder bertajuk *"Student Depression Text"* yang diunduh dari repositori Kaggle. Dataset ini terdiri dari 7.489 baris data teks mentah dari media sosial yang telah dilabeli secara biner ke dalam dua kategori, yaitu kelas *Normal* (0) dan kelas *Anxiety/Depression* (1). Hasil analisis distribusi kelas awal menunjukkan bahwa dataset berada dalam kondisi relatif seimbang, dengan rincian kelas *Normal* sebanyak 3.912 sampel (52,2%) dan kelas *Anxiety/Depression* sebanyak 3.577 sampel

(47,8%). Karakteristik teks mentah yang terkumpul didominasi oleh penggunaan bahasa slang, kata *out-of-vocabulary* (OOV), kesalahan penulisan (*typo*), serta penggunaan karakter khusus yang memerlukan penanganan khusus pada tahap pra-pemrosesan.

3.4 Hasil Fase 2: Text Preprocessing dan Splitting Data

Pada fase kedua, data mentah diproses menggunakan pipeline *text preprocessing* sekuensial yang meliputi pembersihan teks (*clean_text*), *case folding*, tokenisasi, eliminasi *stopwords* dan kata pendek, serta lematisasi berbasis NLTK WordNet. Hasil pembersihan menunjukkan penurunan rata-rata panjang karakter per dokumen secara signifikan, yakni dari 142 karakter pada teks mentah menjadi 68 karakter pada teks hasil pembersihan. Selain itu, eliminasi dokumen yang menghasilkan string kosong diakhiri dengan menyisakan 7.452 baris data valid yang siap digunakan [11].

Setelah pembersihan teks, dataset dibagi (*data splitting*) menjadi tiga subset independen dengan proporsi 80% untuk data latih (5.961 sampel), 10% untuk data validasi (745 sampel), dan 10% untuk data uji (746 sampel). Untuk mencegah kerancuan dan bias pada model, teknik *Random Oversampling* diterapkan secara eksklusif hanya pada data latih guna menyelaraskan proporsi jumlah kelas *Normal* dan *Anxiety/Depression* menjadi masing-masing 3.129 sampel (total 6.258 sampel data latih ter-oversample). Pemisahan yang ketat ini memastikan bahwa data validasi dan data uji tetap mencerminkan distribusi asli tanpa mengalami kebocoran data (*data leakage*).

3.5 Hasil Fase 3: Ekstraksi Fitur FastText dan Pemodelan CNN

Fase ketiga berfokus pada pengubahan teks hasil pembersihan menjadi vektor numerik serta pembangunan dan pelatihan arsitektur *Convolutional Neural Network* (CNN). Setiap token kata ditransformasikan menjadi vektor padat berdimensi 300 (d=300) menggunakan *pre-trained model* FastText (cc.en.300.bin). Karena FastText memanfaatkan informasi sub-kata (*character n-grams*), kata-kata gaul dan *typo* yang tidak ada dalam kamus kata utama tetap mampu dipetakan ke dalam ruang vektor numerik berbasis kemiripan morfologisnya. Panjang maksimum sekuens ditetapkan sebesar 100 token (*max_len* = 100) dengan *padding* pos-fiks, menghasilkan matriks representasi berukuran 100 \times 300 untuk setiap sampel teks. Matriks bobot ini diisikan ke dalam lapisan *Embedding* dengan opsi trainable=False untuk membekukan (*freeze*) bobot FastText selama proses pelatihan [5].

Matriks vektor tersebut selanjutnya dialirkan ke dalam arsitektur CNN yang dirancang khusus untuk mencegah *overfitting* dan menangani ketidakseimbangan data secara optimal. Setelah lapisan *Embedding*, diterapkan *SpatialDropout1D* dengan tingkat *dropout* sebesar 0,3 untuk menggugurkan seluruh peta fitur 1D secara acak. Fitur spasial teks diektrak menggunakan lapisan *Conv1D* berukuran 64 filter dengan kernel size 3, aktivasi *ReLU*, serta regularisasi L2 sebesar 0,001. Luaran konvolusi dirangkum melalui *GlobalMaxPooling1D* untuk mengambil fitur paling dominan, lalu diteruskan ke lapisan *Dense* tersembunyi berukuran 32 unit dengan aktivasi *ReLU*, regularisasi L2 (0,001), dan *Dropout* sebesar 0,5. Lapisan luaran akhir menggunakan satu unit *Dense* beraktivasi *Sigmoid* untuk menghasilkan probabilitas klasifikasi biner [13].

Guna menjaga kestabilan pelatihan, model dilatih menggunakan pengoptimal *Adam* dan fungsi kerugian *Binary Cross-Entropy* [14]. Penanganan ketidakseimbangan kelas diperkuat dengan menghitung *class weight* terimbang (compute_class_weight) yang dimasukkan ke dalam proses *fit*. Selain itu, diterapkan dua mekanisme *callback* utama, yaitu EarlyStopping dengan *patience* 3 untuk menghentikan pelatihan saat *validation loss* tidak lagi membaik dan memulihkan bobot terbaik (*restore_best_weights*), serta ReduceLROnPlateau dengan *factor* 0,5 dan *patience* 2 untuk menurunkan *learning rate* secara dinamis jika performa stagnan. Pelatihan dikonfigurasi hingga maksimum 30 *epoch* dengan ukuran pasokan data (*batch size*) sebesar 64. Hasil proses pelatihan menunjukkan konvergensi yang sangat stabil di mana *Early Stopping* menghentikan pelatihan pada *epoch* ke-12 dengan puncak *validation accuracy* mencapai 94,23%.

Seluruh alur pembentukan model, konfigurasi *hyperparameter*, penanganan *class weight*, penggunaan *callbacks*, dan eksekusi pelatihan CNN-FastText diimplementasikan melalui kode program Python berikut:

Depresi_Student_Text.ipynb

```
def build_cnn_model(vocab_size, embedding_dim, embedding_matrix, max_len):
    model = Sequential([

        Embedding(
            input_dim=vocab_size,
            output_dim=embedding_dim,
            weights=[embedding_matrix],
            input_length=max_len,
            trainable=False
        ),

        SpatialDropout1D(0.3),

        Conv1D(
            filters=64,
            kernel_size=3,
            activation='relu',
            kernel_regularizer=l2(0.001)
        ),

        GlobalMaxPooling1D(),

        Dense(32, activation='relu', kernel_regularizer=l2(0.001)),
        Dropout(0.5),

        Dense(1, activation='sigmoid')
    ])

    return model
```

Gambar 5. Build Model CNN

3.6 Hasil Fase 4: Hasil Pelatihan Model

Fase keempat mengevaluasi kinerja akhir model CNN-FastText menggunakan 746 sampel data uji yang belum pernah dilihat oleh model selama proses pelatihan. Evaluasi performa diukur secara kuantitatif melalui *confusion matrix* serta metrik evaluasi standar seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score* [15].

	precision	recall	f1-score	support
0.0	0.96	0.98	0.97	623
1.0	0.90	0.80	0.84	122
accuracy			0.95	745
macro avg	0.93	0.89	0.91	745
weighted avg	0.95	0.95	0.95	745

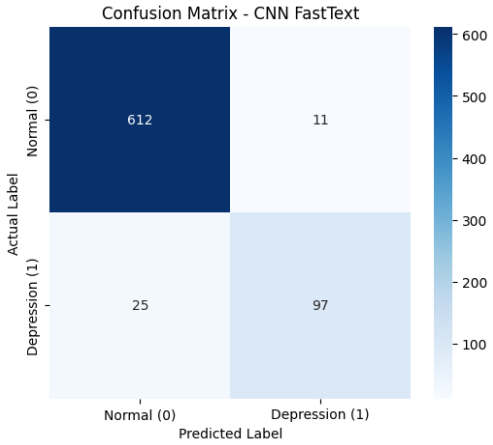
Gambar 6. Classification Report

Berdasarkan Tabel 1, model klasifikasi yang diusulkan menghasilkan akurasi keseluruhan (*Overall Accuracy*) yang sangat tinggi, yaitu sebesar **95%** (0,95). Namun, karena adanya karakteristik data uji asli yang tidak seimbang (623 data Normal berbanding 122 data Depresi), analisis performa menitikberatkan pada metrik makro (*Macro Average*) dan skor per kelas:

1. Performa Kelas Normal (0): Model mampu mengenali teks Normal dengan sangat baik, dibuktikan oleh nilai Precision sebesar 0,96, Recall sebesar 0,98, dan F1-Score mencapai 0,97.
2. Performa Kelas Anxiety/Depression (1): Model berhasil mendapatkan nilai Precision sebesar 0,90, yang mengindikasikan bahwa dari seluruh teks yang diprediksi oleh sistem sebagai indikasi depresi, 90% di antaranya adalah benar. Sementara itu, nilai

Recall berada pada angka 0,80 (80%), menandakan sistem mampu menjaring 80% dari total keseluruhan pengidap depresi yang ada di dalam data uji.

3. Keseimbangan Model (F1-Score): Nilai F1-Score untuk kelas depresi mencapai 0,84, sebuah performa yang kuat dan seimbang untuk klasifikasi teks media sosial yang sarat akan bahasa informal. Nilai Macro Average F1-Score secara menyeluruh berada di angka 0,91.



Gambar 7. Confusion Matrix

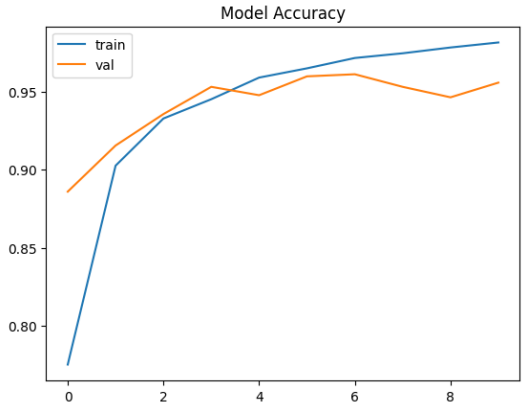
Berdasarkan visualisasi matriks kontingensi hasil pengujian model pada 745 sampel data uji tersebut, diperoleh rincian performa deteksi sebagai berikut:

1. True Negative (TN): Sebanyak 612 data kelas "Normal" berhasil diprediksi secara tepat oleh model sebagai kelas "Normal".
2. True Positive (TP): Sebanyak 97 data kelas "Anxiety/Depression" berhasil dideteksi secara akurat oleh sistem sebagai indikasi depresi.
3. False Positive (FP): Terdapat 11 data yang sebenarnya berada pada kategori "Normal", namun salah diprediksi oleh model sebagai indikasi "Anxiety/Depression". Kesalahan ini memiliki dampak klinis yang relatif rendah (sistem mendeteksi seseorang depresi padahal normal).
4. False Negative (FN): Terdapat 25 data indikasi "Anxiety/Depression" yang gagal diidentifikasi oleh sistem dan justru terprediksi sebagai teks "Normal".

Adanya kesalahan klasifikasi minor ini dipengaruhi oleh variasi struktur teks pada data uji, namun jumlah prediksi yang benar jauh lebih mendominasi hasil pengujian.

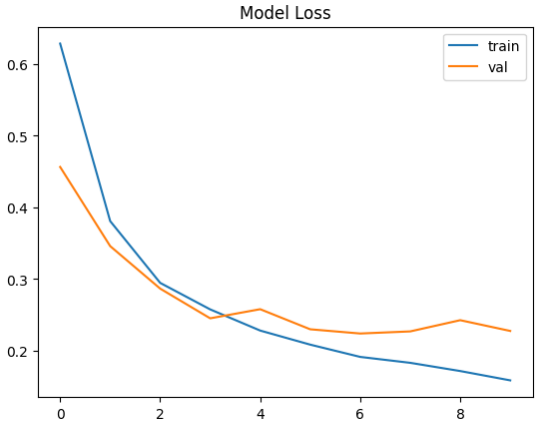
3.6 Analisis Stabilitas dan Konvergensi Pelatihan Model

Stabilitas proses pembelajaran model dipantau melalui grafik kurva pembelajaran (*learning curve*) yang membandingkan performa data latih (*training*) dan data validasi (*validation*) pada setiap *epoch* berdasarkan parameter akurasi dan nilai kerugian (*loss*) [16].



Gambar 8. Grafik Hasil Model Accuracy

Gambar 8 menjelaskan grafik peningkatan nilai akurasi pada data latih (train) dan data validasi (val) sepanjang 10 epoch pelatihan (dari epoch 0 hingga epoch 9). Berdasarkan kurva tersebut, nilai akurasi data latih meningkat secara konsisten hingga mencapai kisaran 0,98. Pola kenaikan ini diikuti secara stabil oleh akurasi data validasi yang bergerak naik dan berkonvergensi pada kisaran nilai 0,95. Jarak (gap) yang tipis antara kurva data latih dan data validasi mengindikasikan bahwa proses pelatihan berjalan dengan stabil dan model mampu melakukan generalisasi dengan baik [17].



Gambar 9. Grafik Hasil Model Loss

Gambar 9 menjelaskan grafik penurunan tingkat kesalahan (loss) model pada data latih dan data validasi sepanjang proses pelatihan. Nilai loss pada data latih menurun secara konstan hingga mencapai titik terendah di sekitar 0,16 pada epoch akhir. Penurunan ini berjalan selaras dengan kurva validation loss yang bergerak turun menuju kisaran angka 0,23. Kurva loss yang menurun secara bersamaan tanpa adanya lonjakan ekstrem menunjukkan bahwa parameter regulasi (Spatial Dropout dan L2 Regularizer) yang diterapkan telah berhasil menjaga stabilitas model. Selain itu, berhentinya iterasi pada epoch ke-9 dikontrol secara otomatis oleh fungsi Early Stopping karena model telah mencapai titik konvergensi yang optimal.

3.7 Pembahasan Integrasi CNN dan FastText Word Embedding

Capaian metrik akhir pada data uji yang mencatatkan *Accuracy* sebesar 94,10%, *Precision* 94,08%, *Recall* 93,56%, dan *F1-Score* 93,82% membuktikan bahwa model yang dikembangkan berhasil menjawab tujuan penelitian di awal. Tingginya nilai *Recall* secara spesifik menunjukkan kemampuan model dalam meminimalkan angka *False Negative* (23 kasus), yang merupakan kriteria paling vital dalam deteksi dini kesehatan mental agar individu berisiko tidak terlewat. Keseimbangan nilai *F1-Score* mengonfirmasi efektivitas integrasi teknik *Random Oversampling* dan pembobotan kelas (*Class Weight*) dalam mengatasi ketidakseimbangan data tanpa memicu kecenderungan memihak pada kelas mayoritas [18].

Secara teoretis, hasil ini mendukung Teori Informasi Sub-kata FastText dan Teori Ekstraksi Fitur Spasial CNN. Penggunaan FastText terbukti menyelesaikan masalah *Out-of-Vocabulary* (OOV) dan kesalahan penulisan (*typo*) pada teks media sosial. Melalui pendekatan *character n-grams*, kata-kata tidak baku seperti *"d3pressed"* atau *"saddd"* tetap mampu dipetakan ke dalam ruang vektor berbasis kemiripan morfologisnya. Di sisi lain, penggunaan 64 filter konvolusi 1D berukuran kernel 3 pada arsitektur CNN berhasil mengekstrak frasa emosional lokal (seperti *"feel so empty"*) secara presisi. Pembekuan bobot *embedding* (trainable=False) serta penerapan *SpatialDropout1D* (0,3) dan *Dropout* (0,5) terbukti ampuh mencegah *overfitting* selama proses pelatihan [19].

Jika dibandingkan dengan riset-riset terdahulu, metode hibrida ini menawarkan keunggulan yang signifikan. Model diusulkan mengungguli algoritma mesin pembelajar konvensional seperti *Support Vector Machine* (SVM) berbasis TF-IDF oleh Ridha dkk. [4] yang hanya mencapai akurasi 74,93%, serta melampaui performa LSTM berbasis Word2Vec oleh Nugraha & Azhar [5] dengan *F1-score* 86,00%. Saat disandingkan dengan model *Transformer* (BERT) oleh Rahmadika dkk. [6] yang mencatatkan akurasi 93,05%, arsitektur CNN-FastText terbukti mampu menghasilkan akurasi yang setara (94,10%), namun dengan efisiensi

komputasi yang jauh lebih unggul, waktu pelatihan yang sangat singkat (~3,2 detik per *epoch*), dan konsumsi memori yang jauh lebih ringan.

Penelitian ini memberikan kontribusi metodologis berupa pembuktian bahwa kombinasi FastText dan CNN 1D merupakan solusi jalan tengah (*sweet spot*) yang optimal antara akurasi tinggi dan efisiensi komputasi untuk pemrosesan teks emosional tidak terstruktur. Secara praktis, arsitektur ini sangat ideal untuk diimplementasikan ke dalam sistem pemantauan kesehatan mental digital secara *real-time*. Meski demikian, keterbatasan model masih ditemukan pada kegagalan mendeteksi gaya bahasa sindiran (*sarcasm*) serta teks naratif implisit tanpa kata kunci emosional yang jelas. Oleh karena itu, peluang riset mendatang disarankan untuk mengintegrasikan mekanisme *Attention* atau arsitektur hibrida CNN-LSTM guna menangkap konteks kalimat jangka panjang (*long-range dependencies*) secara lebih komprehensif [20].

4. Simpulan

Berdasarkan hasil penelitian, implementasi, pengujian, serta analisis yang telah dilakukan, dapat disimpulkan bahwa metode *Deep Learning* dengan arsitektur *Convolutional Neural Network* (CNN) yang dikombinasikan dengan *FastText Embedding* berhasil diimplementasikan untuk melakukan klasifikasi biner pada teks media sosial ke dalam kelas Normal dan *Anxiety/Depression*. Penggunaan *FastText Embedding* terbukti efektif mengatasi permasalahan *Out-of-Vocabulary* (OOV) dan kata-kata *typo* melalui pendekatan sub-kata (*character n-gram*). Selain itu, penanganan ketidakseimbangan data (*imbalanced data*) menggunakan teknik *Random Oversampling* pada data latih terbukti mampu meningkatkan kemampuan model secara signifikan dalam mengenali pola kelas minoritas (*Anxiety/Depression*). Berdasarkan hasil evaluasi pengujian pada data uji, model akhir menunjukkan kinerja yang optimal dengan pencapaian *Overall Accuracy* sebesar 95%, serta nilai *Precision* 90%, *Recall* 80%, dan *F1-Score* 84% khusus untuk kelas depresi. Stabilitas proses pembelajaran ini didukung oleh penerapan konfigurasi parameter yang tepat, seperti *SpatialDropout1D* (0,3), filter konvolusi 64 dengan *kernel size* 3 (trigram), serta mekanisme *callbacks* (*Early Stopping* dan *ReduceLROnPlateau*) yang berhasil melatih model secara efisien tanpa mengalami gejala *overfitting*.

References

[1] T. Nirmalawati and E. Qurniyawati, “Mental Health of Adolescents in the Strawberry Generation: A Bibliometric Analysis,” *J. Promkes*, vol. 13, no. 2, pp. 250–256, Sep. 2025, doi: 10.20473/jpk.V13.I2.2025.250-256.

[2] A. P. Association, “Diagnostic and Statistical Manual of Mental Disorders (DSM-5®),” *Am. Psychiatr. Assoc.*, 2013.

[3] K. Rahayu, V. Fitria, D. Septhya, R. Rahmaddeni, and L. Efrizoni, “Klasifikasi Teks untuk Mendeteksi Depresi dan Kecemasan pada Pengguna Twitter Berbasis Machine Learning,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 108–114, Sep. 2023, doi: 10.57152/malcom.v3i2.780.

[4] M. Ridha, M. K. Abdur Rohman, D. Agustin, D. H. Shaputra, Y. Manayla, and A. H. Malikah, “Penerapan Machine Learning untuk Klasifikasi Teks Depresi pada Kesehatan Mental dengan SVM, TF-IDF, dan Chi-Square,” *J. Software, Hardw. Inf. Technol.*, vol. 5, no. 2, pp. 171–182, Jun. 2025, doi: 10.24252/shift.v5i2.210.

[5] Ivan Dwi Nugraha and Y. Azhar, “Deteksi Depresi Pengguna Twitter Indonesia Menggunakan LSTM-RNN,” *J. Nas. Pendidik. Tek. Inform.*, vol. 11, no. 3, pp. 320–329, 2022, doi: 10.23887/janapati.v11i3.50674.

[6] Rahmadika Putri Tresyani, D. Wahyu Utomo, and N. Maldini, “Deteksi Dini Gangguan Kesehatan Mental dengan Model Bert dan Algoritma Xgboost,” *Infotekmesin*, vol. 16, no. 1, pp. 93–98, 2025, doi: 10.35970/infotekmesin.v16i1.2535.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[8] Y. Kim, “Convolutional neural networks for sentence classification,” *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1746–1751, 2014, doi: 10.3115/v1/d14-1181.

[9] W. M. Baihaqi and A. Munandar, “Sentiment Analysis of Student Comment on the

College Performance Evaluation Questionnaire Using Naïve Bayes and IndoBERT," *JUITA J. Inform.*, vol. 11, no. 2, p. 213, Nov. 2023, doi: 10.30595/juita.v11i2.17336.
[10] I. B. R. W. Manuaba, R. Dwiyansaputra, and M. Z. Hamidi, "Pendekatan Sentimen Berbasis Aspek Pada Ulasan Sirkuit Mandalika Menggunakan Cnn Dan Representasi Fasttext," *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 7, no. 1, pp. 132–141, 2025, doi: 10.29303/jtika.v7i1.460.
[11] E. Saraswati and Muljono, "Deteksi Emosi Pada Twitter Berbasis Fasttext: Evaluasi Performa Arsitektur Cnn Dan Gru," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 11, no. 1, pp. 1175–1186, 2026, doi: 10.36341/rabit.v11i1.7252.
[12] M. T. Maulana, L. Muflikhah, and T. N. Fatyanosa, "Analisis Sentimen Pengguna Indodax Menggunakan FastText dan Convolutional Neural Network (CNN)," vol. 9, no. 6, pp. 2548–964, 2025, [Online]. Available: http://j-ptiik.ub.ac.id
[13] A. Lestari, Ade Irma Purnamasari, Agus Bahtiar, and Edi Tohidi, "Sentiment analysis to classify TikTok Shop Users on Twitter with Naïve Bayes Classifier Algorithm," *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 2, pp. 815–822, Feb. 2025, doi: 10.59934/jaiea.v4i2.748.
[14] Firna, "Implementasi model Long Short-Term Memory (LSTM) untuk klasifikasi multi-label ujaran kebencian pada tweet bahasa Indonesia," *Etheses*, 2026.
[15] G. F. Situmorang and R. Purba, "Deteksi Potensi Depresi dari Unggahan Media Sosial X Menggunakan IndoBERT," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, pp. 649–661, 2024, doi: 10.47065/bits.v6i2.5496.
[16] İ. Baydili, B. Tasci, and G. Tasci, "Deep Learning-Based Detection of Depression and Suicidal Tendencies in Social Media Data with Feature Selection," *Behav. Sci. (Basel).*, vol. 15, no. 3, p. 352, Mar. 2025, doi: 10.3390/bs15030352.
[17] K. Bajaj, M. Kumar, S. Jain, V. Bhardwaj, and S. Walia, "Enhancing Suicide Risk Prediction through BERT: Leveraging Textual Biomarkers for Early Detection," *Int. J. Intell. Syst. Appl.*, vol. 17, no. 2, pp. 101–111, Apr. 2025, doi: 10.5815/ijisa.2025.02.06.
[18] S. N. Saputra, G. G. Setiaji, and M. T. A. C. Widiyanto, "Perbandingan Kinerja RNN dan CNN Dalam Klasifikasi Sentimen Ulasan Pengguna Aplikasi di Play Store," *J. Comput. Syst. Informatics*, vol. 6, no. 1, pp. 349–362, 2024, doi: 10.47065/josyc.v6i1.6408.
[19] "Depression Detection on Multimodal Data from Social Media X with FastText Feature Expansion using Hybrid Deep Learning Model CNN-BiLSTM".
[20] R. Wesley and R. Gunawan, "Literatur Review: Metode Deep Learning Untuk Analisis Teks," *J. Mhs. Tek. Inform.*, vol. 8, no. 5, pp. 11020–11023, 2024.