

Perbandingan Performa Arsitektur *Machine Learning* untuk Deteksi Dini Depresi Berbasis *Natural Language* *Processing* dalam Bahasa Indonesia

Amora Antonio Pangestu^a, Akhmad Rezki Purnajaya^{*}

^aProgram Studi Sistem Informasi, Fakultas Komputer, Universitas Universal, Indonesia

^bProgram Studi Teknik Perangkat Lunak, Fakultas Komputer, Universitas Universal, Indonesia

email: amora250618@gmail.com^a, rezkipurnajaya@gmail.com^b

Abstract. Depression has become an urgent global mental health crisis. The utilization of Natural Language Processing (NLP) in digital text analysis offers great potential for early, non-intrusive detection of depression. This study presents a comparative analysis of three machine learning architectures—Naive Bayes, Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN)—for classifying Indonesian-language texts. The research methodology began with data acquisition and preprocessing (cleaning, case folding, tokenization, and stopword removal) of a dataset containing 10,801 psychologist-annotated texts. Models were trained on 75% of the data and evaluated on the remaining 25% using accuracy, precision, recall, and F1-score metrics. The results show that the CNN architecture achieved the highest performance with a uniform F1-score of 94%. This performance slightly outperformed the LSTM model (93%) and significantly surpassed Naive Bayes (86%). The limitations of this study include its focus on binary classification and the absence of Transformer architectures. These findings provide an important foundation for developing mental health detection systems in Indonesia that are more accurate, adaptive, effective, and relevant to local cultural contexts.

Keywords: Machine Learning, Convolutional Neural Network, Long-Short Term Memory, Naïve Bayes, Natural Language Processing, Mental Health, Depression

Abstrak. Depresi telah menjadi krisis kesehatan mental global yang mendesak. Pemanfaatan Natural Language Processing (NLP) pada analisis teks digital menawarkan potensi besar untuk deteksi dini depresi secara non-intrusif. Penelitian ini menyajikan analisis komparatif dari tiga arsitektur machine learning, yaitu Naive Bayes, Long Short-Term Memory (LSTM), dan Convolutional Neural Network (CNN) untuk mengklasifikasikan teks berbahasa Indonesia. Metodologi penelitian dimulai dengan akuisisi dan pra-pemrosesan (pembersihan, case folding, tokenisasi, stopword removal) dataset 10.801 teks yang teranotasi psikolog. Model dilatih pada 75% data dan

^{*}Corresponding author. E-mail: rezkipurnajaya@gmail.com

Diterima 15 Oktober 2025, Disetujui 27 Oktober 2025, Diterbitkan Oktober 2025

Sitasi IEEE: A. A. Pangestu and A. R. Purnajaya, “Perbandingan Performa Arsitektur Machine Learning untuk Deteksi Dini Depresi Berbasis Natural Language Processing dalam Bahasa Indonesia,” *J-SIGN (Journal of Informatics, Information System, and Artificial Intelligence)*, vol. 3, no. 2, pp. 93–104, 2025.

DOI: 10.24815/j-sign.v3i2.49873

dievaluasi pada 25% data uji menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil menunjukkan bahwa arsitektur CNN mencapai performa tertinggi dengan F1-score seragam sebesar 94%. Kinerja ini sedikit melampaui model LSTM (93%) dan secara signifikan mengungguli Naive Bayes (86%). Keterbatasan penelitian ini mencakup fokus pada klasifikasi biner dan belum digunakannya arsitektur Transformer. Temuan ini memberikan landasan penting untuk pengembangan deteksi kesehatan mental di Indonesia yang lebih akurat, adaptif, efektif, dan relevan terhadap konteks budaya lokal.

Kata Kunci: *machine learning, convolutional neural network, Long-Short Term Memory, Naïve Bayes, Natural Language Processing*, kesehatan mental, depresi

1. Pendahuluan

Depresi merupakan gangguan mental yang telah menjadi salah satu tantangan kesehatan masyarakat paling signifikan di abad ke-21. Secara global, depresi adalah penyebab utama kedua disabilitas [1]. Krisis ini diperburuk oleh pandemi COVID-19 yang dimulai pada tahun 2020. Pandemi tidak hanya memicu krisis kesehatan fisik, tetapi juga resesi ekonomi yang mendalam, yang berujung pada pemutusan hubungan kerja (PHK) massal dan peningkatan stres psikologis [2].

Di Indonesia, skala masalah ini tidak kalah mengkhawatirkan. Menurut data Riset Kesehatan Dasar (Riskesmas) 2018, prevalensi gangguan depresi pada remaja usia 15-24 tahun mencapai 6.2% dan terus meningkat seiring bertambahnya usia, mencapai puncaknya pada kelompok usia 75 tahun ke atas (8.9%) [3]. Angka-angka ini kemungkinan merupakan estimasi yang lebih rendah dari kondisi sebenarnya, mengingat adanya stigma sosial yang kuat. Dampak dari depresi yang tidak tertangani dapat mengarah pada komplikasi serius seperti penurunan produktivitas, isolasi sosial, hingga munculnya pikiran untuk bunuh diri [4].

Ledakan penggunaan media sosial dan platform digital lainnya telah menghasilkan volume data tekstual yang sangat besar [5]. Jejak digital ini, ketika dianalisis, dapat berfungsi sebagai proksi yang kuat untuk memantau tren kesehatan mental. Fenomena ini membuka jalan bagi *pemanfaatan Natural Language Processing* (NLP), sebuah cabang kecerdasan buatan, untuk mengembangkan alat deteksi dini depresi [6]. Metode otomatis ini menawarkan skalabilitas yang tidak dapat ditandingi oleh pendekatan klinis tradisional, namun juga dihadapkan pada tantangan signifikan seperti isu etis privasi data dan potensi bias algoritmik [6].

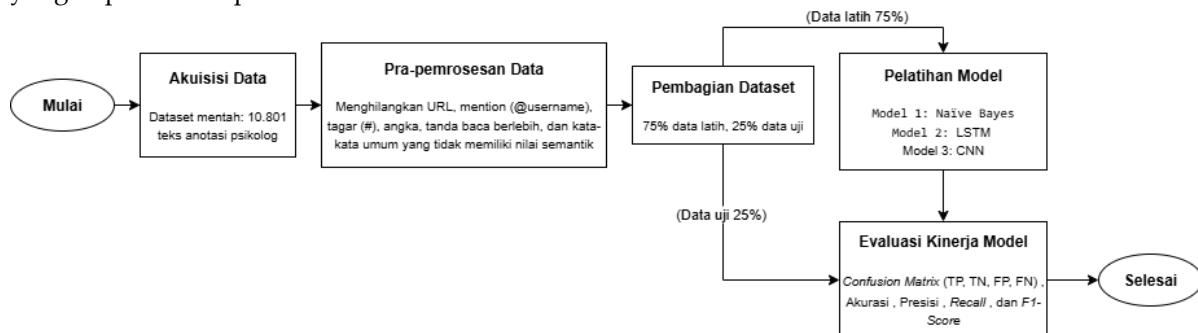
Upaya untuk mendeteksi depresi secara otomatis dari data teks telah berkembang seiring kemajuan dalam bidang *machine learning* dan NLP. Penelitian awal sebagian besar mengandalkan model klasik seperti *Naive Bayes* dan *Support Vector Machines* (SVM) [7]. Kemunculan *deep learning* menandai pergeseran paradigma, di mana arsitektur seperti *Convolutional Neural Networks* (CNN) dan *Long Short-Term Memory* (LSTM) mulai menunjukkan kinerja yang unggul [8]. CNN terbukti efektif menangkap fitur lokal (n-gram), sementara LSTM unggul dalam memodelkan dependensi sekuensial [9]. Saat ini, lanskap NLP didominasi oleh arsitektur berbasis Transformer seperti BERT, yang mampu menangkap pemahaman kontekstual yang sangat mendalam dan telah menetapkan standar kinerja baru [10].

Meskipun terdapat banyak penelitian mengenai deteksi depresi, masih ada celah signifikan dalam literatur, terutama untuk bahasa non-Inggris seperti bahasa Indonesia. Terdapat kekurangan studi yang melakukan perbandingan langsung antara tiga paradigma arsitektur fundamental, yaitu probabilistik (*Naive Bayes*), sekuensial (LSTM), dan konvolusional (CNN) pada dataset depresi berbahasa Indonesia yang berkualitas tinggi. Penelitian ini dirancang untuk mengisi celah tersebut dengan menyediakan

perbandingan performa model kinerja dan menetapkan *baseline* kinerja yang kuat sebagai titik acuan untuk evaluasi arsitektur yang lebih canggih di masa depan.

2. Metodologi Penelitian

Kerangka metodologi penelitian ini dirancang secara sistematis untuk memastikan validitas dan reliabilitas hasil. Prosesnya dimulai dari akuisisi data yang cermat, diikuti oleh pra-pemrosesan teks yang terstruktur, pembagian data latih dan data uji, pemodelan menggunakan tiga arsitektur yang berbeda, dan diakhiri dengan evaluasi kinerja yang komprehensif. Berikut alur kerja proses penelitian yang dapat dilihat pada Gambar 1.



Gambar 1. Flowchart metodologi penelitian

2.1. Akuisisi dan Karakterisasi Dataset

Sumber data merupakan fondasi dari setiap penelitian machine learning. Dalam studi ini, digunakan dataset publik yang bersumber dari repositori Github (*andrebudiman/DatasetIndikasiDepresi/Data10k_Csv.csv*), yang secara khusus dikurasi untuk tugas deteksi indikasi depresi dalam teks berbahasa Indonesia [11]. Dataset ini memiliki beberapa karakteristik kunci yang menjadikannya sangat cocok untuk penelitian ini:

- Ukuran dan Skala: Dataset ini terdiri dari 10.801 data teks, sebuah volume yang cukup besar untuk melatih dan mengevaluasi model *deep learning* secara efektif tanpa mengalami *overfitting* yang parah.
- Anotasi Profesional: Kekuatan utama dari dataset ini terletak pada proses anotasinya. Setiap teks diklasifikasikan oleh psikolog profesional, yang memastikan bahwa label yang diberikan ("Terindikasi Depresi" dan "Tidak Terindikasi Depresi") memiliki validitas klinis. Ini merupakan keunggulan signifikan dibandingkan dataset yang dilabeli berdasarkan kata kunci atau deklarasi diri pengguna, yang seringkali mengandung banyak *noise* dan ketidakakuratan.
- Konteks Spesifik: Data dikumpulkan dalam periode waktu yang terdefinisi, yaitu antara 18 November 2020 hingga 7 Desember 2020. Data ini berasal dari pengguna di pulau Jawa, Bali, dan Sumatera, yang memberikan konteks geografis dan temporal pada temuan penelitian. Pengetahuan ini penting untuk memahami batasan generalisasi dari model yang dihasilkan.
- Skema Klasifikasi: Tugas yang didefinisikan adalah klasifikasi biner, di mana setiap teks harus dikategorikan ke dalam salah satu dari dua kelas: "Terindikasi Depresi" atau "Tidak Terindikasi Depresi". Skema ini jelas dan memungkinkan evaluasi kinerja model yang lugas.

2.2. Pra-pemrosesan dan Standardisasi Teks

Data teks mentah dari internet seringkali tidak terstruktur dan mengandung banyak *noise*. Oleh karena itu, tahap pra-pemrosesan sangat penting untuk mengubah data mentah menjadi format yang bersih dan terstandarisasi yang dapat diproses secara efektif oleh model *machine learning*. Tahapan pra-pemrosesan dalam penelitian ini terdiri dari beberapa langkah berurutan [12]:

- Pembersihan Data (*Data Cleaning*): Langkah pertama adalah menghilangkan karakter dan elemen yang tidak relevan secara semantik. Ini termasuk penghapusan URL, mention pengguna (@username), tagar (#), angka, dan tanda baca yang berlebihan. Proses ini bertujuan untuk mengurangi noise dalam dataset dan memastikan model fokus pada konten tekstual yang bermakna.
- *Case Folding*: Semua teks dikonversi menjadi huruf kecil. Langkah ini penting untuk standardisasi, memastikan bahwa kata-kata seperti "Depresi", "depresi", dan "DEPRESI" diperlakukan sebagai token yang sama oleh model. Hal ini secara efektif mengurangi ukuran kosakata dan mencegah fragmentasi fitur.
- Tokenisasi (*Tokenization*): Kalimat-kalimat dipecah menjadi unit-unit linguistik yang lebih kecil, yaitu kata atau token. Ini adalah langkah fundamental yang memungkinkan analisis teks pada tingkat kata, yang diperlukan untuk sebagian besar model NLP.
- Penghapusan Stopword (*Stopword Removal*): Kata-kata umum yang sering muncul tetapi memiliki sedikit nilai semantik (misalnya, "dan", "di", "yang", "dari") dihilangkan dari teks. Penghapusan stopwords membantu mengurangi dimensi data dan memungkinkan model untuk lebih fokus pada kata-kata yang membawa informasi penting terkait sentimen atau kondisi mental.

Setelah melalui tahapan pra-pemrosesan ini, dataset yang dihasilkan menjadi lebih bersih, lebih efisien untuk diproses, dan lebih kondusif untuk proses analisis dan pemodelan oleh algoritma *machine learning*.

2.3. Arsitektur Naïve Bayes

Naïve Bayes (NB) adalah model klasifikasi probabilistik sederhana namun kuat yang didasarkan pada Teorema Bayes. Teorema Bayes dinyatakan sebagai berikut:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \quad (1)$$

di mana C adalah kelas (label) dan X adalah fitur. Model ini disebut "naïf" karena asumsi utamanya adalah bahwa semua fitur (kata) bersifat independen secara kondisional satu sama lain, mengingat kelasnya. Meskipun asumsi ini jarang benar dalam bahasa alami, NB seringkali berkinerja sangat baik dalam praktik, terutama untuk klasifikasi teks [13].

NB sangat cocok untuk data teks yang direpresentasikan sebagai frekuensi kata (model *bag-of-words*). Untuk mengklasifikasikan dokumen baru, NB menghitung probabilitas *maximum a posteriori* (MAP) untuk setiap kelas, yaitu kelas yang paling mungkin mengingat kata-kata dalam dokumen tersebut :

$$C_{map} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} P(c) \prod_{i=1}^n P(W_i | c) \quad (2)$$

di mana $P(c)$ adalah probabilitas prior kelas c , dan $P(W_i | c)$ adalah probabilitas kondisional dari kata W_i muncul dalam dokumen dari kelas c [13].

Salah satu tantangan dalam NB adalah masalah probabilitas nol. Jika sebuah kata dalam dokumen uji tidak pernah muncul dalam dokumen latih untuk kelas tertentu, probabilitas kondisionalnya akan menjadi nol, yang akan membuat seluruh probabilitas posterior menjadi nol. Untuk mengatasi ini, digunakan teknik penghalusan (*smoothing*), seperti *add-one smoothing* (atau *Laplace smoothing*), yang menambahkan satu ke setiap hitungan kata. Ini memastikan bahwa tidak ada probabilitas yang bernilai nol [14].

2.4. Arsitektur Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) adalah jenis *Recurrent Neural Network* (RNN) yang dirancang khusus untuk mengatasi masalah dependensi jangka panjang dalam data sekuensial, seperti teks. Tidak seperti RNN standar yang rentan terhadap masalah *vanishing gradient*, LSTM menggunakan arsitektur sel yang canggih untuk mengatur aliran informasi [15].

Inti dari arsitektur LSTM adalah status sel (*cell state*), yang berfungsi sebagai memori atau "sabuk konveyor" yang membawa informasi relevan di sepanjang urutan. Aliran informasi ini diatur oleh tiga "gerbang" (*gates*):

- *Forget Gate*: Gerbang ini memutuskan informasi mana dari *cell state* sebelumnya yang harus dibuang atau dilupakan. Gerbang ini menggunakan fungsi sigmoid untuk menghasilkan nilai antara 0 ("lupakan sepenuhnya") dan 1 ("pertahankan sepenuhnya") berdasarkan input saat ini dan *hidden state* sebelumnya.
- *Input Gate*: Gerbang ini menentukan informasi baru mana yang relevan untuk disimpan dalam *cell state*. Proses ini melibatkan dua bagian: fungsi *sigmoid* yang memutuskan nilai mana yang akan diperbarui, dan fungsi *tanh* yang membuat vektor nilai kandidat baru.
- *Output Gate*: Gerbang ini menentukan apa yang akan menjadi output (*hidden state*) berikutnya. *Hidden state* ini adalah versi yang telah difilter dari *cell state*, yang diputuskan oleh fungsi *sigmoid* dan kemudian dilewatkan melalui fungsi *tanh*.

Untuk tugas klasifikasi teks, arsitektur tipikal terdiri dari *Embedding Layer* yang mengubah kata menjadi vektor, diikuti oleh satu atau lebih lapisan LSTM yang memproses urutan vektor kata, dan diakhiri dengan *Dense Layer* dengan fungsi aktivasi *softmax* atau *sigmoid* untuk menghasilkan probabilitas kelas [15].

2.5. Arsitektur Convolutional Neural Network (CNN)

Meskipun awalnya dirancang untuk pemrosesan gambar, *Convolutional Neural Networks* (CNN) telah berhasil diadaptasi untuk tugas-tugas NLP. Dalam konteks teks, CNN memperlakukan kalimat sebagai matriks 1D dari vektor kata (*word embeddings*). Arsitektur CNN untuk klasifikasi teks biasanya terdiri dari lapisan-lapisan berikut [16]:

- *Embedding Layer*: Sama seperti pada LSTM, lapisan ini mengubah setiap kata dalam kalimat menjadi representasi vektor padat. Hasilnya adalah sebuah matriks di mana baris mewakili kata dan kolom mewakili dimensi vektor.
- *Convolutional Layers*: Lapisan ini adalah inti dari CNN. Alih-alih filter 2D seperti pada gambar, CNN untuk teks menggunakan filter 1D yang meluncur di sepanjang urutan kata. Filter-filter ini, dengan ukuran yang bervariasi (misalnya, ukuran 2, 3, 4 kata), berfungsi sebagai detektor *n-gram*. Setiap

filter belajar untuk mengenali pola linguistik lokal yang spesifik (misalnya, frasa seperti "merasa lelah", "tidak ada harapan") yang sangat indikatif terhadap sentimen atau kondisi mental.

- *Pooling Layers*: Setelah konvolusi, biasanya diterapkan lapisan *max-pooling*. Lapisan ini mengurangi dimensi dari *feature maps* dengan hanya mempertahankan fitur yang paling menonjol (nilai maksimum) yang terdeteksi oleh setiap filter. Ini membantu model menjadi invarian terhadap posisi fitur dan mengurangi kompleksitas komputasi.
- *Fully Connected Layer*: Fitur-fitur yang diekstraksi dari lapisan pooling kemudian diratakan (*flattened*) dan dihubungkan ke lapisan *fully connected*, yang berfungsi sebagai pengklasifikasi. Lapisan ini menggabungkan semua fitur yang telah dipelajari untuk membuat prediksi akhir.

2.6. *Tensorflow*

TensorFlow merupakan sebuah kerangka kerja (*framework*) komputasi simbolik dan pustaka perangkat lunak (*software library*) sumber terbuka (*open-source*) yang dikembangkan oleh tim Google Brain. Fondasi utamanya dirancang untuk komputasi numerik berkinerja tinggi, yang menjadikannya platform dominan untuk melatih dan menerapkan model *Machine Learning* (ML) dan *Deep Learning* (DL) [17].

Nama "*TensorFlow*" sendiri merujuk pada dua konsep fundamental yang mendasari arsitekturnya [17]:

- *Tensor*: Representasi data dalam *TensorFlow* adalah melalui *tensor*, yaitu struktur data multidimensional yang merupakan generalisasi dari skalar (*tensor rank-0*), vektor (*tensor rank-1*), dan matriks (*tensor rank-2*). Seluruh data, mulai dari input, parameter model, hingga output, diabstraksikan sebagai *tensor*.
- *Flow*: Konsep ini mengacu pada aliran (*flow*) data (*tensor*) melalui serangkaian operasi matematika. Operasi-operasi ini direpresentasikan dalam sebuah graf komputasi (*computational graph*) statis. Dalam graf ini, simpul (*nodes*) melambangkan operasi matematika (misalnya, adisi, multiplikasi matriks, atau fungsi aktivasi), sementara rusuk (*edges*) merepresentasikan *tensor* yang mengalir di antara simpul-simpul tersebut.

Fleksibilitas *TensorFlow* memungkinkannya diterapkan dalam domain rekayasa yang kompleks yaitu *Natural Language Processing* (NLP), sebagai tugas-tugas sekuensial, arsitektur seperti *Recurrent Neural Networks* (RNNs) dan *Long Short-Term Memory* (LSTM). *TensorFlow* juga menjadi platform utama untuk implementasi arsitektur Transformer, yang mendasari model *sequence-to-sequence* untuk terjemahan mesin dan model bahasa skala besar lainnya [17].

2.7. *Metrik Evaluasi Performa*

Untuk memastikan evaluasi yang objektif terhadap kemampuan generalisasi model, dataset dibagi dengan rasio 75% untuk data latih (*training data*) dan 25% untuk data uji (*testing data*). Model dilatih hanya menggunakan data latih, dan kinerjanya dievaluasi pada data uji yang belum pernah dilihat sebelumnya.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2. Penilaian *confusion matrix*

Kinerja setiap model dievaluasi menggunakan *Confusion Matrix* seperti yang terlihat pada Gambar 2, sebuah tabel yang memvisualisasikan kinerja algoritma klasifikasi. Matriks ini terdiri dari empat komponen utama [18]:

- *True Positive* (TP): Jumlah sampel positif ("Terindikasi Depresi") yang diprediksi dengan benar.
- *True Negative* (TN): Jumlah sampel negatif ("Tidak Terindikasi Depresi") yang diprediksi dengan benar.
- *False Positive* (FP): Jumlah sampel negatif yang salah diprediksi sebagai positif.
- *False Negative* (FN): Jumlah sampel positif yang salah diprediksi sebagai negatif.

Dari komponen-komponen ini, beberapa metrik evaluasi kuantitatif dihitung. Dalam konteks diagnosis medis seperti deteksi depresi, setiap metrik memiliki interpretasi yang penting [18]:

- *Accuracy* (Akurasi): Mengukur proporsi prediksi yang benar secara keseluruhan. Rumusnya adalah $(TP+TN)/(TP+TN+FP+FN)$. Meskipun umum digunakan, akurasi bisa menyesatkan pada dataset yang tidak seimbang.
- *Precision* (Presisi): Mengukur proporsi prediksi positif yang benar-benar positif. Rumusnya adalah $TP/(TP+FP)$. Presisi tinggi penting untuk meminimalkan false positive, yaitu salah melabeli individu sehat sebagai depresi, yang dapat menyebabkan kecemasan yang tidak perlu.
- *Recall* (Sensitivitas): Mengukur proporsi kasus positif aktual yang berhasil diidentifikasi oleh model. Rumusnya adalah $TP/(TP+FN)$. *Recall* tinggi sangat krusial dalam konteks ini untuk meminimalkan false negative, yaitu gagal mendeteksi individu yang sebenarnya mengalami depresi, yang dapat menunda intervensi penting.
- *F1-Score*: Merupakan rata-rata harmonik dari presisi dan *recall*. Rumusnya adalah $2 * (Presisi * Recall) / (Presisi + Recall)$. *F1-score* memberikan ukuran kinerja yang seimbang, sangat berguna ketika biaya dari false positive dan false negative sama-sama signifikan.
- *Macro Average* (rata-rata makro): menilai rata-rata evaluasi metrik dari *Precision*, *Recall*, dan *F1-score* secara terpisah untuk setiap kelas tanpa mempertimbangkan jumlah sampel.
- *Weighted Average* (rata-rata bobot): sama seperti *Macro Average*, yang bedanya *Weighted Average* mempertimbangkan jumlah sample dalam menilai rata-rata.

3. Hasil dan Pembahasan

Bagian ini menyajikan hasil eksperimen secara rinci, dimulai dari analisis pelatihan model deep learning, diikuti oleh evaluasi kinerja pada data uji menggunakan *confusion matrix*, dan diakhiri dengan analisis komparatif terpadu dari ketiga model yang diuji.

3.1. Dinamika Pelatihan Model

Proses pelatihan dari tiga jenis model yang berbeda memberikan wawasan awal tentang bagaimana setiap arsitektur belajar dari data. Tabel 1 menunjukkan perbandingan kinerja pelatihan pada epoch pertama dan epoch ke-10 untuk model *deep learning*, serta hasil pelatihan untuk *Naïve Bayes*.

Tabel 1. Hasil pelatihan ketiga model

	Epoch	Accuracy	Loss	val_acc	val_loss
CNN	1	0.7041	0.5598	0.8663	0.3503
	10	0.9967	0.0199	0.9437	0.2018
LSTM	1	0.7424	0.4977	0.8739	0.3155
	10	0.9950	0.0155	0.9322	0.3515
NB	1	0.8560	-	-	-

Dari Tabel 1, terlihat bahwa akurasi validasi (val_acc), yang merupakan metrik penting untuk kemampuan generalisasi, menunjukkan bahwa CNN (94.37%) sedikit lebih unggul dari LSTM (93.22%) pada akhir pelatihan. Selain itu, *validation loss* (val_loss) pada model CNN juga lebih rendah (0.2018) dibandingkan LSTM (0.3515), yang mengindikasikan bahwa model CNN mungkin sedikit lebih baik dalam menggeneralisasi pengetahuannya ke data baru. Pendekatan *Naïve Bayes* yang non-iteratif menghasilkan akurasi sebesar 85.60% pada data latih.

3.2. Evaluasi Kinerja pada Data Uji

Evaluasi akhir dilakukan pada 25% data uji (total 2.521 data) untuk mengukur kinerja prediksi model. Pada kinerja Model *Convolutional Neural Network* (CNN) menunjukkan kinerja yang sangat kuat dan seimbang pada data uji. Hasil lengkapnya disajikan pada Tabel 2. Nilai rata-rata terbobot (*weighted average*) untuk presisi, *recall*, dan *F1-score* semuanya mencapai 94%, menandakan performa yang sangat konsisten di kedua kelas.

Tabel 2. Evaluasi Performa Model CNN

Kelas	Precision	Recall	F1-score	Data test
Terindikasi depresi	93%	92%	93%	1118
Tidak terindikasi	94%	95%	94%	1403
Macro Avg	94%	94%	94%	2521
Weighted Avg	94%	94%	94%	2521

Model LSTM juga menunjukkan kinerja yang sangat baik, meskipun sedikit di bawah CNN, dengan hasil yang dirangkum pada Tabel 3. Model ini menunjukkan nilai rata-rata terbobot untuk presisi, *recall*, dan *F1-score* yang seragam di angka 93%.

Tabel 3. *Confusion Matrix* LSTM

Kelas	Precision	Recall	F1-score	Data test
Terindikasi depresi	91%	92%	92%	1118
Tidak terindikasi	94%	93%	93%	1403
Macro Avg	93%	93%	93%	2521
Weighted Avg	93%	93%	93%	2521

Model *Naïve Bayes* menunjukkan kinerja yang lebih rendah dibandingkan dengan dua model *deep learning*. Hasilnya dapat dilihat pada Tabel 4. Terdapat ketidakseimbangan yang cukup besar antara presisi (80%) dan *recall* (90%) untuk kelas "Terindikasi Depresi", yang mengindikasikan bahwa model ini cenderung menghasilkan lebih banyak *false positive* meskipun berhasil menangkap sebagian besar kasus depresi yang sebenarnya.

Tabel 4. *Confusion Matrix* Naïve Bayes

Kelas	Precision	Recall	F1-score	Data test
Terindikasi depresi	80%	90%	85%	1118
Tidak terindikasi	91%	82%	86%	1403
Macro Avg	86%	86%	86%	2521
Weighted Avg	86%	86%	86%	2521

Keunggulan tipis namun konsisten dari CNN atas LSTM dapat diinterpretasikan melalui perbedaan fundamental dalam cara kedua arsitektur memproses teks. Kekuatan inti CNN terletak pada kemampuannya sebagai detektor fitur lokal yang efisien dalam mengidentifikasi *n-gram* (frasa pendek) yang signifikan, terlepas dari posisinya dalam kalimat. Sebaliknya, LSTM dirancang untuk memahami urutan dan dependensi jangka panjang. Keunggulan CNN menunjukkan bahwa penanda linguistik depresi dalam dataset ini mungkin lebih bersifat lokal dan berbasis frasa (misalnya, "aku capek hidup", "tidak ada harapan"). Kemampuan CNN untuk mengenali pola-pola ini tampaknya menjadi strategi yang sedikit lebih efektif daripada pemodelan sekuensial holistik oleh LSTM.

3.3. Analisis Komparatif Terpadu

Untuk memfasilitasi perbandingan langsung, kinerja ketiga model pada data uji dirangkum dalam Tabel 4. Metrik yang ditampilkan adalah nilai rata-rata terbobot (*weighted average*) untuk memberikan gambaran kinerja yang seimbang di kedua kelas (terindikasi depresi dan tidak terindikasi depresi).

Tabel 4. Perbandingan Kinerja Terpadu Model Klasifikasi

Model	Akurasi	Presisi (Weighted Avg)	Recall (Weighted Avg)	F1-Score (Weighted Avg)
Convolutional Neural Network (CNN)	94%	94%	94%	94%
Long Short-Term Memory (LSTM)	93%	93%	93%	93%
Naïve Bayes (NB)	86%	86%	86%	86%

Model CNN secara konsisten unggul 1% di atas LSTM di semua metrik utama. Meskipun perbedaan ini kecil, konsistensinya di seluruh metrik dan didukung oleh akurasi validasi yang lebih tinggi selama pelatihan menunjukkan keunggulan yang nyata, meskipun tipis, dari arsitektur konvolusional untuk tugas ini. Kesenjangan kinerja antara model *deep learning* (CNN dan LSTM) dan model *machine learning* klasik (*Naïve Bayes*) sangat signifikan. CNN mengungguli *Naïve Bayes* sebesar 8% pada *F1-score*. Hal ini dengan kuat menunjukkan bahwa kemampuan model *deep learning* untuk secara otomatis belajar representasi fitur yang kompleks dari teks memberikan keuntungan besar dibandingkan dengan pendekatan *bag-of-words* probabilistik yang lebih sederhana.

3.4. Diskusi

Hasil eksperimen yang telah disajikan memberikan dasar yang kuat untuk diskusi yang lebih mendalam mengenai implikasi teoretis dan praktis dari temuan ini. Salah satu temuan paling menarik dari penelitian ini adalah keunggulan tipis namun konsisten dari CNN atas LSTM. Perbedaan kinerja ini, meskipun hanya 1%, dapat diinterpretasikan melalui perbedaan fundamental dalam cara kedua arsitektur ini memproses teks. LSTM dirancang untuk memahami urutan dan dependensi jangka panjang dalam sebuah kalimat. Kekuatannya terletak pada kemampuannya untuk memodelkan struktur gramatikal dan konteks yang mengalir dari awal hingga akhir teks. Di sisi lain, kekuatan inti CNN terletak pada kemampuannya untuk bertindak sebagai detektor fitur lokal yang hierarkis. Dengan menggunakan filter berukuran berbeda, CNN secara efisien mengidentifikasi *n-gram* (frasa pendek) yang signifikan secara statistik, terlepas dari di mana frasa tersebut muncul dalam kalimat.

Keunggulan CNN dalam konteks ini menunjukkan bahwa penanda linguistik depresi dalam dataset ini mungkin lebih bersifat lokal dan berbasis frasa daripada bergantung pada struktur kalimat yang kompleks. Bahasa yang digunakan oleh individu yang mengalami depresi seringkali ditandai oleh penggunaan frasa-frasa kunci yang kuat dan berulang, seperti "aku capek hidup", "ingin mati saja", "merasa hampa", atau "tidak ada harapan". Kemampuan CNN untuk belajar mengenali pola-pola *n-gram* yang kuat ini dan menganggapnya sebagai sinyal penting, terlepas dari konteks gramatikal yang lebih luas, tampaknya menjadi strategi yang sedikit lebih efektif daripada pemodelan sekuensial holistik yang dilakukan oleh LSTM. Dengan kata lain, untuk tugas ini, kehadiran frasa-frasa indikatif yang kuat mungkin merupakan prediktor yang lebih kuat daripada *urutan* kata-kata dalam kalimat secara keseluruhan.

Sementara itu, kinerja *Naïve Bayes* yang jauh lebih rendah mengkonfirmasi keterbatasan pendekatan *bag-of-words* untuk tugas yang bernuansa seperti ini. Dengan mengabaikan urutan kata dan konteks sepenuhnya, *Naïve Bayes* kehilangan informasi semantik yang krusial. Model ini hanya dapat mengandalkan frekuensi kemunculan kata-kata secara individual, yang terbukti tidak cukup untuk membedakan secara andal antara ekspresi kesedihan biasa dan bahasa yang secara klinis mengindikasikan depresi.

4. Kesimpulan

Penelitian ini telah melakukan evaluasi komparatif yang sistematis terhadap tiga arsitektur *machine learning* yang berbeda untuk tugas deteksi depresi dari teks berbahasa Indonesia. Temuan utama menunjukkan bahwa *Convolutional Neural Network* (CNN) adalah model yang paling efektif di antara ketiganya, mencapai *F1-score* sebesar 94%. Kinerja ini sedikit melampaui *Long Short-Term Memory* (LSTM)

dengan *F1-score* 93% dan secara signifikan lebih baik daripada *Naïve Bayes* dengan *F1-score* 86%. Studi ini memberikan dua kontribusi utama: pertama, menetapkan *baseline* kinerja yang kuat untuk tugas ini pada dataset berbahasa Indonesia yang dianotasi secara klinis. Kedua, memberikan wawasan bahwa ekstraksi fitur lokal melalui deteksi *n-gram*, yang merupakan kekuatan CNN, mungkin merupakan strategi yang lebih efektif untuk mengidentifikasi penanda linguistik depresi dibandingkan dengan pemodelan konteks sekuensial murni.

Berdasarkan temuan dan keterbatasan studi ini, beberapa arah yang menjanjikan untuk penelitian di masa depan dapat direkomendasikan:

1. Adopsi Model Transformer: Langkah paling mendesak adalah melakukan *benchmarking* model *state-of-the-art* seperti IndoBERT pada dataset ini. Diharapkan model ini akan memberikan peningkatan akurasi yang signifikan dan menetapkan standar kinerja baru.
2. Klasifikasi Tingkat Keparahan: Penelitian selanjutnya harus bergerak melampaui klasifikasi biner ke arah klasifikasi multi-kelas untuk memprediksi tingkat keparahan depresi. Ini akan sangat meningkatkan nilai klinis dari model tersebut.
3. Analisis Multimodal: Teks hanyalah salah satu modalitas komunikasi. Mengintegrasikan analisis teks dengan modalitas lain seperti audio dan video dapat menghasilkan model deteksi yang lebih holistik dan akurat.
4. Studi Longitudinal: Alih-alih hanya mendeteksi status depresi pada satu titik waktu, penelitian longitudinal yang melacak bahasa pengguna dari waktu ke waktu dapat membantu mengidentifikasi pola yang mengarah pada *onset* depresi, memungkinkan intervensi preventif.
5. Pengembangan Aplikasi Klinis: Menerjemahkan model-model ini menjadi aplikasi nyata, seperti *chatbot* pendukung atau dasbor untuk para klinisi.

Daftar Pustaka

- [1] GBD 2019 Mental Disorders Collaborators, "Global, regional, and national burden of 12 mental disorders in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019," *The Lancet Psychiatry*, vol. 9, no. 2, pp. 137–150, 2022.
- [2] J. M. Cénat *et al.*, "Prevalence of symptoms of depression, anxiety, insomnia, posttraumatic stress disorder, and psychological distress among populations affected by the COVID-19 pandemic: A systematic review and meta-analysis," *Psychiatry Research*, vol. 295, p. 113599, 2021.
- [3] S. Idaiani *et al.*, "Individual and area-level factors associated with depression in Indonesia: a multilevel analysis using the 2018 national basic health research," *BMC Public Health*, vol. 25, no. 1, p. 3298, 2025.
- [4] W. Marx *et al.*, "Major depressive disorder," *Nature Reviews Disease Primers*, vol. 9, no. 1, p. 44, 2023.
- [5] D. Liu *et al.*, "Detecting and measuring depression on social media using a machine learning approach: systematic review," *JMIR Mental Health*, vol. 9, no. 3, p. e27244, 2022.
- [6] B. G. Teferra *et al.*, "Screening for depression using natural language processing: literature review," *Interactive Journal of Medical Research*, vol. 13, no. 1, p. e55067, 2024.
- [7] A. Bucur *et al.*, "On the state of NLP approaches to modeling depression in social media: A post-COVID-19 outlook," *IEEE Journal of Biomedical and Health Informatics*, 2025.
- [8] O. O. Adebajji *et al.*, "Adaptation of transformer-based models for depression detection," *Computación y Sistemas*, vol. 28, no. 1, pp. 151–165, 2024.
- [9] S. Aisyiyah and W. Maharani, "Analisis berbasis emosional pada depresi di media sosial menggunakan pendekatan Convolutional Neural Network," *eProceedings of Engineering*, vol. 10, no. 2, 2023.
- [10] K. Hasan and J. Saquer, "A comparative analysis of Transformer and LSTM models for detecting suicidal ideation on Reddit," in *Proc. 2024 Int. Conf. on Machine Learning and Applications (ICMLA)*, IEEE, 2024.
- [11] A. Budiman, *DatasetIndikasiDepresi* [Dataset], GitHub, 2023. [Online]. Available: <https://github.com/andrebudiman/DatasetIndikasiDepresi>. [Accessed: Jul. 1, 2025].

- [12] A. R. Purnajaya, O. Noverio, Very, and C. Alvaro, "Analisis ancaman COVID-19 varian XBB di Indonesia pada jejaring media sosial Twitter menggunakan text mining," *JITTER: Jurnal Ilmiah Teknologi dan Komputer*, vol. 4, no. 1, 2023.
- [13] D. N. Chandra, G. Indrawan, and I. N. Sukaraja, "Klasifikasi berita lokal Radar Malang menggunakan metode Naïve Bayes dengan fitur N-Gram," *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 10, no. 1, pp. 11–19, 2016.
- [14] Willianto *et al.*, "Implementasi teori Naive Bayes dalam klasifikasi ujaran kebencian di Facebook," *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 4, pp. 666–671, 2021.
- [15] R. G. W. Sahwahita Hutomo, "Implementasi pemeliharaan prediktif mesin printing di UMKM Batik X dengan metode Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU)," *Thesis*, Universitas Islam Indonesia, 2025.
- [16] A. Jacovi, O. S. Shalom, and Y. Goldberg, "Understanding convolutional neural networks for text classification," *arXiv preprint*, arXiv:1809.08037, 2018.
- [17] D. Novaliendry *et al.*, "Development of a semantic text classification mobile application using TensorFlow Lite and Firebase ML Kit," *Journal Européen des Systèmes Automatisés*, vol. 57, no. 6, p. 1603, 2024.
- [18] F. Gunawan, P. Limuel, and A. R. Purnajaya, "Implementasi Support Vector Machine dan Radial Basis Function untuk klasifikasi makanan vegetarian menggunakan data image," *Journal of Digital Ecosystem for Natural Sustainability*, vol. 3, no. 1, pp. 5–8, 2023.