

**IMPLEMENTASI ALGORITMA LSTM DAN INDOBERTWEET
EMBEDDING UNTUK DETEKSI INDIKASI DEPRESI
BERDASARKAN POSTINGAN MEDIA SOSIAL**

SKRIPSI

EDBERT

211402004



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

**IMPLEMENTASI ALGORITMA LSTM DAN INDOBERTWEET EMBEDDING
UNTUK DETEKSI INDIKASI DEPRESI BERDASARKAN
POSTINGAN MEDIA SOSIAL**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

EDBERT

211402004



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

PERSETUJUAN

Judul : Implementasi Algoritma LSTM Dan IndoBertweet Embedding Untuk Deteksi Indikasi Depresi Berdasarkan Postingan Media Sosial
 Kategori : Skripsi
 Nama Mahasiswa : Edbert
 Nomor Induk Mahasiswa : 211402004
 Program Studi : Sarjana (S-1) Teknologi Informasi
 Fakultas : Ilmu Komputer Dan Teknologi Informasi Sumatera Utara

Medan, 10 Januari 2025

Komisi Pembimbing:

Pembimbing 2,



Dr. Erna Budhiarti Nababan, M.I.T

NIP. 196210262017042001

Pembimbing 1,



Dedy Arisandi, S.T., M.Kom

NIP. 197908312009121002

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua,



Dedy Arisandi, S.T., M.Kom

NIP. 197908312009121002

PERNYATAAN

IMPLEMENTASI ALGORITMA LSTM DAN INDOBERTWEET EMBEDDING
UNTUK DETEKSI INDIKASI DEPRESI BERDASARKAN
POSTINGAN MEDIA SOSIAL

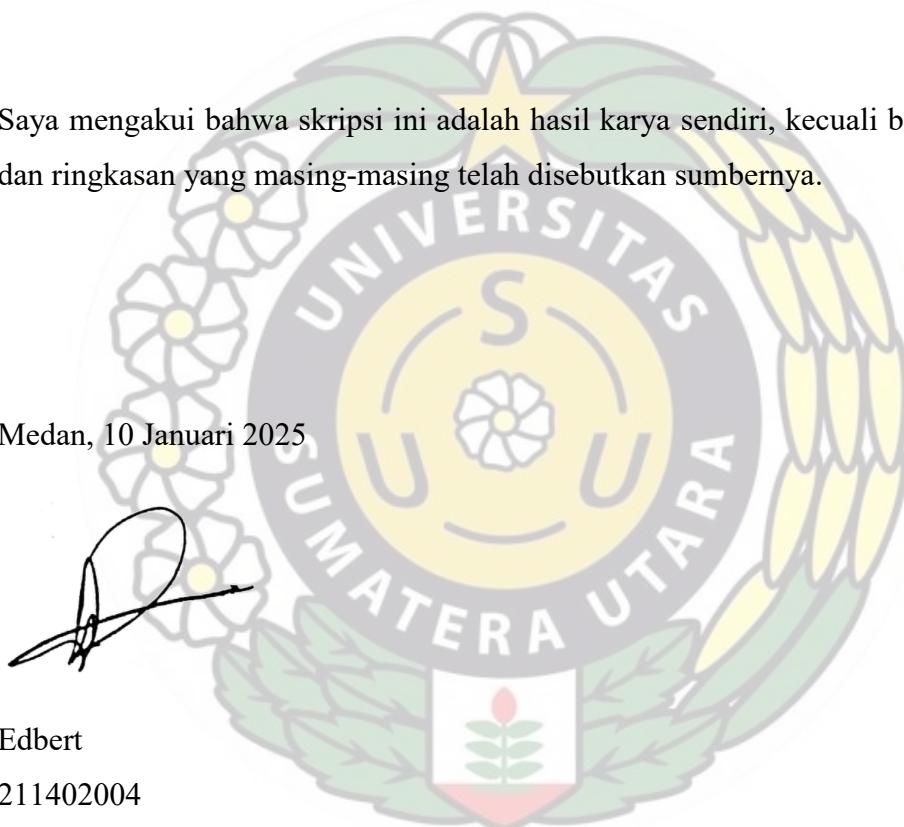
SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 10 Januari 2025

Edbert

211402004

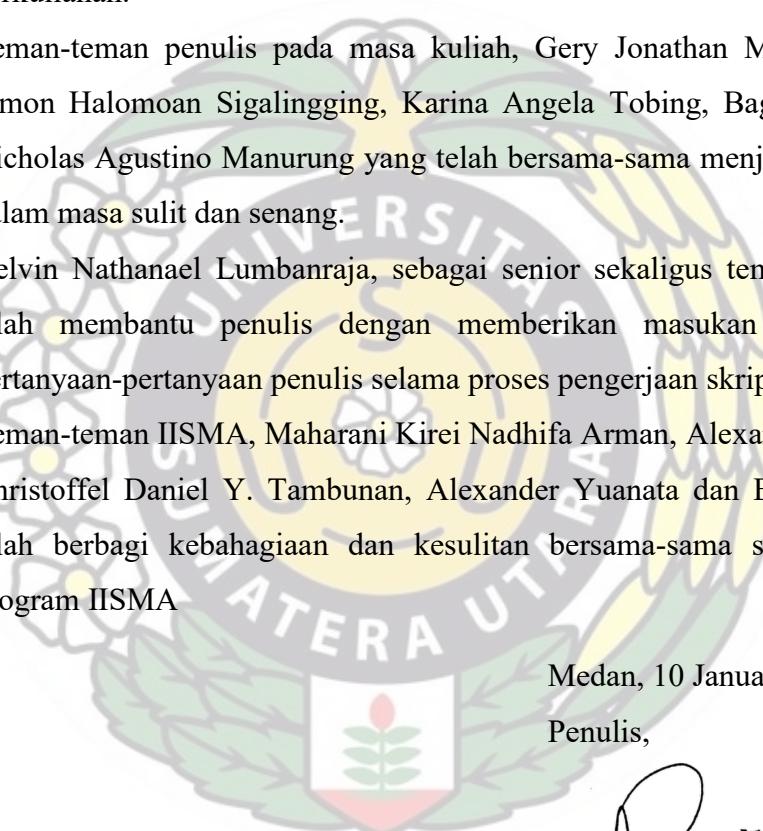


UCAPAN TERIMAKASIH

Penulis mengungkapkan rasa syukur dan terima kasih kepada Tuhan Yang Maha Esa yang telah menngaruniakan berkat-Nya dalam membantu penulis menyelesaikan tugas akhir berupa skripsi sebagai persyaratan dalam meraih gelar sarjana dalam bidang Teknologi Informasi di Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara. Judul skripsi ini adalah “Implementasi Algoritma LSTM dan IndoBERTweet Embedding untuk Deteksi Indikasi Depresi Berdasarkan Postingan Media Sosial”. Penyelesaian skripsi ini terwujud dengan kontribusi dan bantuan dari pihak-pihak yang terlibat dalam perjalanan penggerjaan skripsi penulis. Oleh sebab itu, penulis hendak menyampaikan ucapan terima kasih yang tulus kepada:

1. Keluarga penulis, Ayahanda Makmur dan Ibunda Ida, serta Abangda Leonardo yang selalu memberikan dukungan, semangat, dan doa kepada penulis agar tetap kuat dari awal perjalanan perkuliahan hingga tahap penyelesaian skripsi.
2. Bapak Dedy Arisandi S.T., M.Kom., selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara dan sekaligus sebagai Dosen Pembimbing 1 yang telah memberikan bimbingan, arahan, serta motivasi kepada penulis selama penggerjaan skripsi ini.
3. Ibu Dr. Erna Budhiarti Nababan M.IT., selaku Dosen Pembimbing 2 yang juga memberikan motivasi serta membimbing penulis selama penggerjaan skripsi ini.
4. Bapak Ivan Jaya S.Si., M.Kom., selaku Dosen Penguji 1 yang telah memberikan saran terhadap skripsi ini.
5. Ibu Ade Sarah Huzaifah S.Kom., M.Kom., selaku Dosen Penguji 2 yang juga telah memberikan masukkan untuk meningkatkan skripsi ini.
6. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara
7. Bapak dan Ibu Dosen di Program Studi Teknologi Informasi, Universitas Sumatera Utara, yang telah memberikan pengetahuan yang berlimpah kepada penulis selama masa perkuliahan.
8. Seluruh staff serta pegawai Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, yang telah membantu segala urusan administrasi selama masa perkuliahan.

9. Sahabat-sahabat penulis, Clement Rusli, Clifferd, Derry Setiady, Vincent Aulia, Caroline Wijaya, Sharon Ivana dan Kenrix Tanvano yang saling mendukung dan menemani satu sama lain dalam menjalani kehidupan sejak masa SMP hingga sekarang.
10. Teman-teman Kom A yang telah menyemangati penulis dalam menyelesaikan skripsi ini.
11. Muhammad Ilham dan Kenzie Fubrianto, yang selalu membantu dan memberikan dukungan ketika penulis menghadapi permasalahan selama masa perkuliahan.
12. Teman-teman penulis pada masa kuliah, Gery Jonathan Manurung, Paulus Simon Halomoan Sigalingging, Karina Angela Tobing, Bagus Sadewo, dan Nicholas Agustino Manurung yang telah bersama-sama menjalani perkuliahan dalam masa sulit dan senang.
13. Kelvin Nathanael Lumbanraja, sebagai senior sekaligus teman penulis yang telah membantu penulis dengan memberikan masukan dan menjawab pertanyaan-pertanyaan penulis selama proses penggeraan skripsi
14. Teman-teman IISMA, Maharani Kirei Nadhifa Arman, Alexandra Adjani Lega, Christoffel Daniel Y. Tambunan, Alexander Yuanata dan Erick Lorus yang telah berbagi kebahagiaan dan kesulitan bersama-sama selama menjalani program IISMA



Medan, 10 Januari 2025

Penulis,



Edbert
211402004

ABSTRAK

Depresi merupakan gangguan mental yang ditandai dengan kesedihan yang mendalam dan berkelanjutan, perasaan tidak berharga, bersalah, dan kehilangan minat dalam kehidupan sehari-hari. Media sosial menjadi wadah bagi para remaja atau dewasa muda untuk menceritakan keluh kesah ataupun masalah yang membuat mereka depresi. Akan tetapi, mendeteksi postingan yang terindikasi depresi pada media sosial secara manual bersifat subjektif dalam memahami konteks setiap postingan. Hal ini menjadi pemicu dibutuhkannya suatu pendekatan yang efektif untuk mendeteksi indikasi depresi pada postingan di media sosial. Pada penelitian ini, algoritma LSTM dikombinasikan dengan IndoBERTweet *Embedding* untuk mendeteksi indikasi depresi pada postingan teks media sosial berbahasa Indonesia. Data yang digunakan untuk melatih dan mengevaluasi model sebanyak 12703 data yang diambil dari postingan Twitter (X). Hasil penelitian menunjukkan bahwa sistem yang dibangun dengan algoritma LSTM dan IndoBERTweet *Embedding* mampu mendeteksi indikasi depresi dengan akurasi sebesar 95.8%

Kata Kunci : Depresi, *Long Short-Term Memory* (LSTM), IndoBERTweet *Embedding*

*IMPLEMENTATION OF LSTM ALGORITHM AND INDOBERTWEET
EMBEDDING FOR DEPRESSION INDICATION DETECTION
BASED ON SOCIAL MEDIA POSTS*

ABSTRACT

Depression is a mental disorder marked by profound and enduring sadness, feelings of worthlessness, guilt, and loss of interest in daily life. Social media is a place for teenagers or young adults to share their grievances or problems that make them depressed. However, manually detecting posts that indicate depression on social media is subjective in understanding the context of each post. This motivates the need for a robust method to detect indications of depression in social media posts. In this research, the LSTM algorithm is combined with IndoBERTweet Embedding to detect indications of depression in Indonesian-language social media posts. The data used to train and evaluate the model comprises 12703 data taken from Twitter (X) posts. The results show that the system built with the LSTM algorithm and IndoBERTweet Embedding can detect indications of depression with an accuracy of 95.8%.

Keywords: Depression, Long Short-Term Memory (LSTM), IndoBERTweet Embedding

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN	iv
UCAPAN TERIMAKASIH	v
ABSTRAK	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	6
2.1 Postingan Terindikasi Depresi	6
2.2 Text Preprocessing	7
2.3 Word Embedding	7
2.4 Long Short-Term Memory (LSTM)	8
2.5 Confusion Matrix	11
2.6 Penelitian Terdahulu	12
2.7 Perbedaan Penelitian	17
BAB 3 ANALISIS DAN PERANCANGAN SISTEM	18
3.1 Data yang digunakan	18
3.2 Arsitektur Umum	19
3.2.1 Preprocessing	21
3.2.2 <i>Split Dataset</i>	28
3.2.3 <i>Word Embedding</i>	28
3.2.4 <i>Long Short-Term Memory (LSTM)</i>	29
3.2.5 Output	34
3.3 Perancangan Sistem	34

3.3.1 Desain Halaman Training	35
3.3.2 Desain Halaman Testing	35
3.3.3 Desain Halaman Cek Depresi Teks	36
3.3.4 Desain Halaman Cek Depresi File	37
3.3.5 Desain Chrome Extension	38
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM	40
4.1 Implementasi Sistem	40
4.1.1 Spesifikasi Perangkat Keras dan Lunak	40
4.1.2 Penerapan sistem berbasis Website dan <i>Google Chrome extension</i>	40
4.2 Hasil Pelatihan Model	46
4.3 Pengujian Kinerja Model	49
4.4 Hasil Evaluasi Pakar	59
BAB 5 KESIMPULAN DAN SARAN	62
5.1 Kesimpulan	62
5.2 Saran	62
DAFTAR PUSTAKA	64



DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	14
Tabel 3.1 Dataset Postingan Teks Normal dan Berindikasi Depresi	18
Tabel 3.2 Pemecahan Dataset	19
Tabel 3.3 Implementasi <i>Text Cleaning</i> tanpa emoji	23
Tabel 3.4 Implementasi <i>Text Cleaning</i> dengan emoji	23
Tabel 3.5 Implementasi <i>Text Normalization</i>	24
Tabel 3.6 Implementasi <i>Case Folding</i>	25
Tabel 3.7 Implementasi <i>Stopword Removal</i>	26
Tabel 3.8 Implementasi <i>Tokenization</i>	27
Tabel 3.9 Pengubahan Token menjadi ID	28
Tabel 4.1 Hasil Pengujian Kombinasi Layers	47
Tabel 4.2 Hasil Pengujian Kinerja Model	49
Tabel 4.3 Postingan yang salah dideteksi oleh model dalam testing	51
Tabel 4.4 Perhitungan data yang tidak terindikasi depresi tanpa emoji	56
Tabel 4.5 Perhitungan data yang terindikasi depresi tanpa emoji	56
Tabel 4.6 Hasil Komputasi Nilai Evaluasi tanpa emoji	57
Tabel 4.7 Perhitungan data yang tidak terindikasi depresi menggunakan emoji	58
Tabel 4.8 Perhitungan data yang terindikasi depresi menggunakan emoji	58
Tabel 4.9 Hasil Komputasi Nilai Evaluasi menggunakan emoji	59
Tabel 4.10 Tabel Pertanyaan pada evaluasi pakar	59
Tabel 4.11 Hasil dari evaluasi pakar	60

DAFTAR GAMBAR

Gambar 2.1 Contoh Postingan Depresi pada Twitter (X)	6
Gambar 2.2 Arsitektur LSTM	9
Gambar 2.3 <i>Confusion Matrix</i> dalam matrix 2x2	11
Gambar 3.1 Arsitektur Umum	21
Gambar 3.2 Pseudocode <i>Text Cleaning</i>	22
Gambar 3.3 Pseudocode <i>Text Normalization</i>	24
Gambar 3.4 Pseudocode <i>Case Folding</i>	25
Gambar 3.5 Pseudocode <i>Stopword Removal</i>	26
Gambar 3.6 Pseudocode <i>Tokenization</i>	27
Gambar 3.7 Hasil <i>Word Embedding</i>	29
Gambar 3.8 Desain Antarmuka Halaman Training	35
Gambar 3.9 Desain Antarmuka Halaman Testing	36
Gambar 3.10 Desain Antarmuka Halaman Cek Indikasi Depresi Teks	37
Gambar 3.11 Desain Antarmuka Halaman Cek Indikasi Depresi File	38
Gambar 3.12 Desain Antarmuka <i>Google Chrome Extension</i>	39
Gambar 4.1 Halaman Training	41
Gambar 4.2 Hasil Training	41
Gambar 4.3 Halaman Testing	42
Gambar 4.4 Hasil Testing	42
Gambar 4.5 Evaluasi Model menggunakan <i>Confusion Matrix</i>	43
Gambar 4.6 Halaman Cek Indikasi Depresi File	43
Gambar 4.7 Hasil postingan tidak terindikasi depresi	44
Gambar 4.8 Hasil postingan terindikasi depresi	44
Gambar 4.9 Halaman Cek Indikasi Depresi File	45
Gambar 4.10 Hasil Cek Indikasi Depresi File	45
Gambar 4.11 <i>Google Chrome Extention</i> Deteksi Indikasi Depresi	46
Gambar 4.12 Hasil Deteksi Indikasi Depresi <i>Google Chrome Extension</i>	47
Gambar 4.13 <i>Confusion Matrix</i> Hasil Pengujian Kinerja Model tanpa emoji	55
Gambar 4.14 <i>Confusion Matrix</i> Hasil Pengujian Kinerja Model menggunakan emoji	57

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Depresi adalah gangguan mental yang ditandai oleh perasaan sedih yang mendalam dan berkepanjangan, perasaan tidak berharga, bersalah, dan kehilangan minat dalam kehidupan sehari-hari. Depresi dapat mempengaruhi fisik dan interaksi sosial seseorang. Gejala depresi sering mencakup mudah tersinggung, tekanan emosional, kecemasan, kurangnya semangat, suasana hati yang murung, kehilangan kesenangan, merasa bersalah, gangguan pola tidur dan nafsu makan yang kurang, serta rendahnya tingkat energi. Dampak negatifnya bisa berupa kesulitan berkonsentrasi dan munculnya pikiran negatif. (Riziana & Darmawan, 2023)

Menurut laporan Kementerian Kesehatan, angka penduduk yang berumur 15 tahun ke atas yang mengalami depresi pada tahun 2018 adalah 6.2% dan pada tahun 2023, angka ini hanya menurun 0.1%. Tidak hanya itu, berdasarkan laporan tersebut, hanya 10.4% penderita depresi yang melakukan pengobatan. Penderita depresi yang tidak diobati dapat mengakibatkan tindakan drastis seperti penggunaan narkoba ataupun bunuh diri.

Ada beberapa teknik mendiagnosis depresi, seperti *clinical interview method* dan *self-report questionnaires*. Kedua teknik tersebut memiliki kekurangan masing-masing yaitu tingkat akurasi yang tidak pasti dengan kemungkinan kesalahan mendiagnosa akibat gejala yang mirip dengan penyakit mental lainnya (Smith et al., 2013). Selain itu, subjektivitas dan kemampuan pasien untuk mendeskripsikan gejala sangat berpengaruh terhadap hasil akhir diagnosis yang menjadi salah satu kesulitan dalam melakukan deteksi depresi. (Nordgaard et al., 2013)

Media sosial telah menjadi konteks utama dalam perkembangan remaja dan dewasa muda, dengan mayoritas dari mereka menggunakannya untuk berinteraksi dan berbagi pengalaman hidup sehari-hari. Penggunaan media sosial dapat memiliki pengaruh yang beragam terhadap kesejahteraan psikologis, tergantung pada aktivitas

yang dilakukan, motif penggunaan, dan siapa mitra komunikasi mereka. Studi menunjukkan bahwa salah satu pemicu penggunaan media sosial adalah sebagai tempat yang aman untuk menceritakan hal-hal yang sedang dialami dengan menjaga anonimitas pengguna (Yang et al., 2021). Hal yang diceritakan pada media sosial sangat beragam tergantung penggunanya, seperti konten edukasi, konten liburan, ataupun konten kondisi pengguna. Konten ini bisa berupa konten yang mengandung unsur depresi. Oleh karena itu, indikasi depresi pada postingan media sosial perlu dideteksi secara dini agar dapat ditolong ke tahap selanjutnya.

Teknologi *Machine Learning* dan *Deep Learning* bisa membantu dalam mendeteksi depresi. Penelitian yang menggunakan *Long Short-Term Memory* (LSTM) demi mendeteksi depresi dilakukan oleh Singh et al. (2022). Dataset yang digunakan berasal dari hasil scraping menggunakan *PushShift API* untuk mendapatkan postingan dari Reddit yang berhubungan dengan kesehatan mental dari beberapa *subreddit*. Penelitian ini melakukan *preprocessing* berupa *tokenization*, *data cleaning*, dan *stemming*. Kemudian, data tersebut diubah menjadi vektor menggunakan GloVe *embedding*. Akurasi yang didapatkan menggunakan algoritma LSTM dan GloVe *embedding* adalah 94.88%.

Penelitian yang menggunakan *Random Forest Classifier*, *XGBoost*, dan *Support Vector Machine* untuk melakukan diagnosa depresi yang bisa membedakan dengan PTSD (*Post-Traumatic Stress Disorder*) dilakukan oleh Lorenzoni et al. (2024) yang berjudul. Dataset yang digunakan berasal dari *Distress Analysis Interview Corpus - Wizard of Oz* yang berisi data audio, video, dan transkrip dari interview yang dilakukan. Akurasi paling tinggi dihasilkan oleh model *Random Forest Classifier* dan *XGBoost* dengan nilai akurasi 83.8%.

Word embedding yang digunakan dalam mendeteksi depresi tidak hanya terbatas pada GloVe tetapi juga bisa menggunakan salah satu *contextualized word embedding* yaitu BERT seperti penelitian oleh Devika et al. (2023). Penelitian ini melakukan *preprocessing* berupa *tokenization*, *removing stop words*, dan *part-of-speech tagging*. Lalu, data tersebut diubah menjadi vektor menggunakan BERT *embedding* dan menggunakan algoritma Bi-LSTM untuk melakukan prediksi. Dataset yang digunakan berasal dari media sosial Reddit. Akurasi yang didapatkan ketika menggunakan Bi-LSTM adalah 81%, sedangkan BERT model menghasilkan akurasi sebesar 72%.

Penelitian lain yang menggunakan BERT dalam mendeteksi depresi dilakukan Hidayat & Maharani (2022) yang berjudul “*General Depression Detection Analysis Using IndoBERT Method*”. Penelitian ini menggunakan data yang dikumpulkan dari Twitter (X). Selanjutnya, dilakukan *preprocessing data* agar data dapat digunakan untuk training pada IndoBERT model. Akurasi yang didapatkan masih pada angka 51%. Hal ini disebabkan oleh beberapa kata yang masih belum bisa diproses oleh IndoBERT seperti ‘wkwkwk’ dan kata-kata diluar Bahasa Indonesia.

IndoBERTweet adalah model bahasa yang dirancang khusus untuk menangani teks Twitter berbahasa Indonesia dengan lebih baik dibandingkan IndoBERT. Model ini dibangun dengan mengadaptasi IndoBERT, model BERT monolingual yang terlatih dengan data dari Wikipedia, artikel berita, dan korpus web Indonesia, dengan menambahkan kosakata yang spesifik untuk domain media sosial. Kelebihan utama IndoBERTweet adalah kemampuannya untuk menangkap nuansa bahasa informal, singkatan, dan emotikon yang sering digunakan dalam postingan media sosial, yang membuatnya sangat cocok untuk mendeteksi indikasi depresi pada teks Twitter. Adaptasi kosa kata domain menggunakan rata-rata embedding subword memungkinkan model untuk mempelajari konteks domain dengan efisiensi yang tinggi dan performa yang lebih baik, sehingga mengungguli IndoBERT dalam tugas berbasis media sosial (Koto et al., 2021). Hal ini menjadikan IndoBERTweet lebih andal dalam memahami ekspresi emosi dan bahasa informal yang sering ditemukan pada teks di platform media sosial.

Berdasarkan penelitian yang telah dilakukan, penulis melakukan penerapan IndoBERTweet *Embedding* dan algoritma *Long Short-Term Memory* untuk mendeteksi postingan media sosial yang memiliki indikasi depresi. Penulis memberikan penelitian ini judul “*Implementasi Algoritma LSTM dan IndoBERTweet Embedding untuk Deteksi Indikasi Depresi Berdasarkan Postingan Media Sosial*”.

1.2 Rumusan Masalah

Depresi merupakan masalah kesehatan mental yang sangat serius dan perlu dideteksi serta dicegah secara dini. Teknik manual seperti interview atau kuesioner yang digunakan pakar masih belum bisa mengdiagnosis individu secara akurat. Banyak individu yang mengalami depresi tidak mengungkapkan masalah mereka secara terbuka, melainkan memilih untuk mencerahkan perasaan dan kondisi mental mereka

melalui media sosial seperti Twitter (X). Postingan di media sosial yang berhubungan dengan depresi sering kali mengandung ungkapan tentang masalah pribadi seperti *bullying*, kekerasan, tekanan/stres, masalah keluarga, dan lain-lain. Namun, mendeteksi postingan yang mengindikasikan depresi pada media sosial secara manual bersifat subjektif dalam memahami konteks setiap postingan. Hal ini menjadi pemicu dibutuhkannya suatu pendekatan yang efektif untuk mendeteksi indikasi depresi pada postingan di media sosial.

1.3 Tujuan Penelitian

Untuk mendeteksi indikasi depresi terhadap postingan media sosial dengan mengimplementasikan algoritma LSTM dan IndoBERTweet *Embedding*.

1.4 Batasan Masalah

Ada beberapa batasan masalah yang ditetapkan dalam penelitian ini, yaitu:

1. Data yang digunakan merupakan data teks yang dimuat pada file berekstensi .csv
2. Data postingan media sosial yang digunakan berasal dari Twitter dan berbahasa Indonesia.
3. Gambar, video link URL, maupun angka tidak digunakan dalam proses deteksi.
4. Model yang dihasilkan akan diimplementasikan dalam sistem berbasis website dan *google chrome extention*

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Mempelajari kinerja dari algoritma LSTM dan IndoBERTweet *Embedding* dalam mendeteksi depresi pada postingan media sosial berbahasa Indonesia.
2. Membantu lembaga-lembaga ataupun instansi seperti dinas kesehatan, pakar, dan masyarakat dalam mendeteksi indikasi depresi pada postingan media sosial.
3. Menjadi sumber referensi untuk penelitian-penelitian selanjutnya dalam mengembangkan teknik mendeteksi depresi.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini dibagi menjadi lima bagian yaitu:

Bab 1: Pendahuluan

Bagian ini mencakup informasi terkait dengan penelitian ini, yaitu latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab 2: Landasan Teori

Bagian landasan teori mencakup teori-teori yang perlu dipahami guna menambah pemahaman dari penelitian ini seperti postingan depresi, algoritma LSTM, IndoBERTweet *Embedding* serta penelitian-penelitian terdahulu yang dijadikan sebagai referensi pada penelitian ini.

Bab 3: Analisis dan Perancangan Sistem

Bab ini mencakup pembahasan arsitektur umum penelitian, tahap *preprocessing data*, perancangan sistem yang akan dibangun, serta model yang akan dikembangkan menggunakan algoritma LSTM.

Bab 4: Implementasi dan Pengujian Sistem

Pada bagian ini, tahap-tahap implementasi dan penerapan desain yang telah dirancang akan dijelaskan secara rinci. Selain itu, evaluasi pada sistem juga dipaparkan pada bagian ini

Bab 5: Kesimpulan dan Saran

Pada bagian kesimpulan dan saran, dijelaskan rangkuman dari penelitian yang dilakukan serta saran untuk penyempurnaan penelitian selanjutnya.

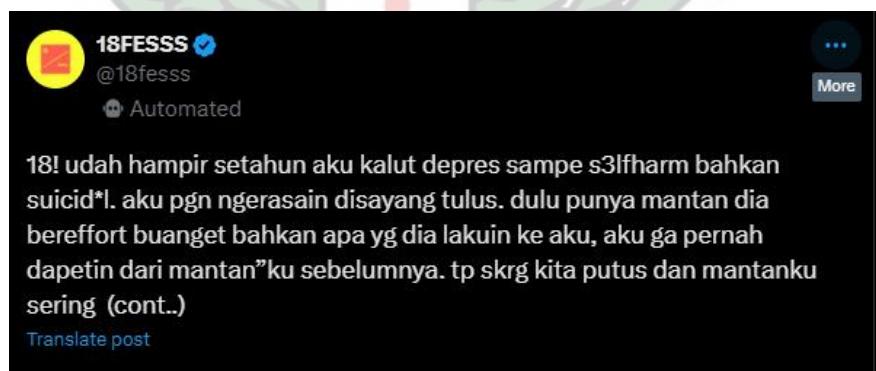
BAB 2

LANDASAN TEORI

2.1 Postingan Terindikasi Depresi

Mendiagnosa seseorang untuk depresi sering kali menjadi tantangan bagi para pakar psikologi dikarenakan perilaku dan gejala yang berbeda-beda. Dalam lingkungan tertentu, seseorang tidak merasa bebas untuk mengutarakan apa masalah yang dihadapi secara terbuka karena kritikan yang mungkin didapatkan. Menggunakan media sosial menjadi solusi untuk menjaga privasi dan anonimitas yang memberikan rasa aman bagi seseorang. (El-Ramly et al., 2021)

Media sosial merupakan salah satu sarana penting dalam menjalankan komunikasi atau mendapatkan berita dalam dunia maya. Media sosial menyediakan forum yang berhubungan dengan kesehatan mental sebagai wadah untuk pengguna menceritakan kondisi kesehatan yang sedang dirasakan saat ini. Kata-kata pada postingan media sosial yang mengindikasikan depresi dari user ditandai dengan penggunaan kata depresi, sedih, lelah, capek, menangis, gelisah, tidak ada harapan dan lainnya (Leis et al., 2020; Sikström et al., 2023) . Contoh postingan yang mengindikasikan depresi pada media sosial Twitter (X) dipaparkan pada Gambar 2.1.



Gambar 2.1 Contoh Postingan Depresi pada Twitter (X)

Gambar di atas merupakan salah satu contoh postingan tentang masalah pribadi seseorang yang mengakibatkan pengguna tersebut mengalami depresi dan menceritakan pengalamannya pada media sosial. Penggunaan kata “depres” merupakan salah satu indikasi yang menyatakan bahwa pengguna tersebut sedang mengalami depresi.

2.2 Text Preprocessing

Pada tahap *text preprocessing*, dilakukan pengolahan data ke dalam bentuk yang lebih terstruktur. *Text preprocessing* dilakukan untuk memperoleh inti data yang terdapat pada dataset (Kadhim, 2018). *Text preprocessing* terdiri dari beberapa tahap seperti *text cleaning*, *text normalization*, *case folding*, *stopword removal*, dan *tokenization*.

2.3 Word Embedding

Word embedding adalah teknik representasi kata menjadi multi dimensi vektor angka dengan mencari relasi semantik dari setiap kata berdasarkan kesamaan makna (Elnagar et al., 2023). Teknik ini diperlukan agar memudahkan model dalam memproses data berupa vektor angka daripada data teks. *Word embedding* dikategorikan menjadi tiga, yaitu *Traditional*, *Static*, dan *Contextualized word embedding* (Birunda & Devi, 2020).

Traditional word embedding merepresentasikan kata berdasarkan frekuensi dari setiap kata yang terdapat pada dataset. Contoh dari *traditional word embedding* adalah TF-IDF dan *co-occurrence matrix*. *Static word embedding* merepresentasikan kata berdasarkan probabilitas dari setiap kata tanpa memandang konteks dan diubah menjadi *dense vectors*. Contoh dari *static word embedding* adalah *Word2Vec*, *GloVe*, and *FastText* (Birunda & Devi, 2020). *Contextualized word embedding* merepresentasikan kata berdasarkan makna kata dari konteks yang berbeda-beda sehingga dua kata yang sama bisa memiliki token yang berbeda jika makna dari kata tersebut berbeda secara konteks (Elnagar et al., 2023). Contoh dari *contextualized word embedding* adalah *ELMo* dan *BERT*.

Bidirectional Encoder Representations from Transformers, atau dikenal sebagai BERT, merupakan *word embedding* yang mampu mempelajari konteks relasi antar kata berdasarkan makna dari kata tersebut (Birunda & Devi, 2020). BERT menggunakan *encoder* untuk *tokenized* setiap kata dari sebuah kalimat dan

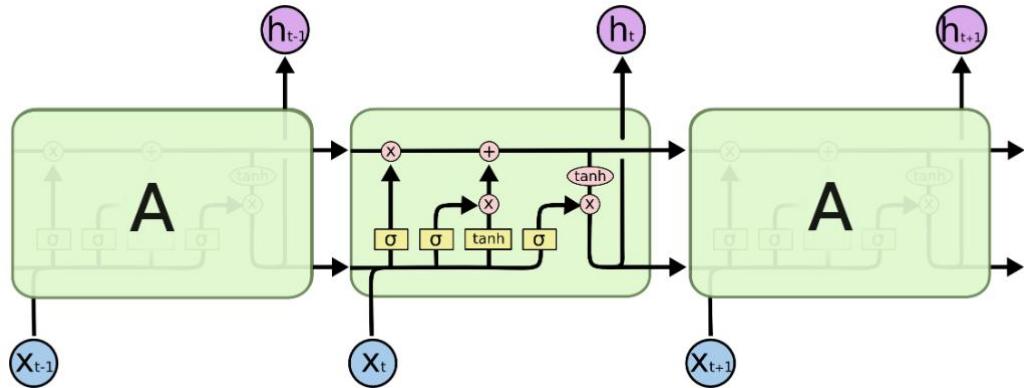
direpresentasikan menjadi vektor berdasarkan makna kata sesuai dengan konteks dari kalimat. Misalnya kata “python” pada kalimat Saya baru belajar bahasa Python dan Python dapat ditemukan di Afrika dan Asia akan dipisahkan menjadi dua vektor yang berbeda menggunakan BERT (Bello et al., 2023).

Model BERT memiliki beberapa turunan model termasuk model yang telah melakukan *training* dengan data bahasa Indonesia. Model ini bernama IndoBERTweet, yang merupakan turunan dari model IndoBERT dengan melakukan *training* pada dataset postingan Twitter (X) dalam bahasa Indonesia berjumlah 26 juta tweet yang dikumpulkan dalam kurun waktu satu tahun. Model IndoBERTweet dilatih agar dapat mengenali pola dan karakteristik penggunaan bahasa yang sering digunakan pada Twitter (X) (Koto et al., 2021).

Kelebihan dari model IndoBERTweet adalah model ini mampu memahami konteks dalam bahasa Indonesia, terutama kosakata yang sering ditemukan pada media sosial (Koto et al., 2021). Hal ini dikarenakan IndoBERTweet menggantikan kosakata dari model sebelumnya yaitu IndoBERT dengan kosakata baru yang didapatkan melalui dataset corpus Twitter (X) berbahasa Indonesia. Kosakata baru yang terdapat pada model IndoBERTweet sebesar 46% dari total 31.984 jenis kata yang ada pada *vocabulary* IndoBERTweet (Koto et al., 2021).

2.4 Long Short-Term Memory (LSTM)

Long short-term memory adalah turunan *Recurrent Neural Network* (RNN) yang dimodifikasi guna menyelesaikan permasalahan dalam mempertahankan memori jangka panjang (Manaswi, 2018). Sebuah *memory cell* LSTM memiliki *output gate*, *input gate*, dan sebuah *forget gate*. *Input gate* berperan dalam mengontrol jumlah data yang masuk ke dalam sebuah *cell*, sedangkan *forget gate* mengatur jumlah *values* di dalam sebuah *cell* yang akan menjadi *output* dari LSTM (Salman et al., 2021).



Gambar 2.2 Arsitektur LSTM (Sumber: colah.github.io)

Gambar 2.2 menggambarkan perulangan dari *cell* pada LSTM. Proses awal LSTM adalah menentukan informasi yang akan disimpan dan akan dibuang pada *forget gate*. Penentuan ini berdasarkan nilai h_{t-1} dan x_t yang menghasilkan nilai f_t pada persamaan 2.1. Nilai f_t bernilai antara 0 dan 1, dimana 1 berarti informasi akan disimpan sedangkan nilai 0 berarti informasi akan dibuang. Nilai f_t akan disimpan pada vektor C_{t-1} .

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

Keterangan:

f_t : *Forget Gate*

W_f : Berat gerbang neuron pada *forget gate*

h_{t-1} : *Hidden state*

x_t : *Input*

b_f : Bias pada *input gate*

σ : Fungsi sigmoid

Langkah selanjutnya adalah menentukan informasi mana yang akan disimpan pada *memory cell*. Proses ini melibatkan *input gate* dan sebuah tanh *layer*. *Input gate* menentukan informasi mana yang akan ditambahkan, kemudian vektor baru yaitu \tilde{C}_t akan dibentuk menggunakan fungsi tanh untuk menampung informasi yang akan ditambahkan menggunakan persamaan 2.3

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh (W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.3)$$

Keterangan:

i_t	: Input gate
W_i	: Berat gerbang neuron pada <i>input gate</i>
b_i	: Bias pada <i>input gate</i>
\tilde{C}_t	: <i>Cell state</i> baru sementara
W_C	: Berat gerbang neuron pada <i>cell state</i>
b_C	: Bias pada <i>cell state</i>

C_{t-1} akan diubah menjadi C_t menggunakan persamaan 2.4 yaitu dengan perkalian C_{t-1} dengan f_t ditambah dengan i_t dikali \tilde{C}_t .

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.4)$$

Keterangan:

C_t	: <i>Cell state</i> baru
f_t	: <i>Forget Gate</i>
C_{t-1}	: <i>Cell state</i> lama
i_t	: <i>Input gate</i>
\tilde{C}_t	: <i>Cell state</i> baru sementara

Langkah terakhir adalah menentukan *output* dari LSTM berdasarkan C_t yang telah didapatkan sebelumnya dari persamaan 2.4 dengan menggunakan sigmoid *layer* dan tanh. Sigmoid *layer* digunakan untuk menentukan bagian dari C_t yang akan digunakan dan tanh memastikan agar nilai berada diantara -1 dan 1. Lalu, dilakukan perkalian terhadap *output* dari sigmoid *gate* untuk menampilkan *output* yang telah ditentukan.

$$o_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh (C_t) \quad (2.6)$$

Keterangan:

o_t	: <i>Output gate</i>
W_o	: Berat gerbang neuron pada <i>output gate</i>
b_o	: Bias pada <i>output gate</i>
h_t	: Hasil akhir yang telah ditentukan
C_t	: <i>Cell state</i> baru

2.5 Confusion Matrix

Confusion Matrix adalah konsep pada *Machine Learning* ataupun *Deep Learning* untuk melakukan visualisasi terhadap kinerja model dalam melakukan klasifikasi (Caelen, 2017). *Confusion Matrix* berisi data mengenai hasil klasifikasi aktual dan prediksi yang telah dilakukan dan dihasilkan oleh model (Santra & Christy, 2012).

		Actually Positive (1)	Actually Negative (0)
		True Positives (TPs)	False Positives (FPs)
Predicted Positive (1)			
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)	

Gambar 2.3 *Confusion Matrix* dalam matrix 2x2

Confusion Matrix memiliki empat karakteristik yang digunakan untuk perhitungan umum yaitu (P. Singh et al., 2021):

1. *True Positive* (TP) merupakan data positif yang diprediksikan positif
2. *True Negative* (TN) merupakan data negatif yang diprediksikan negatif
3. *False Positive* (FP) merupakan data negatif yang diprediksikan positif
4. *False Negative* (FN) merupakan data positif yang diprediksikan negatif

Perhitungan umum yang dapat dilakukan dengan informasi dari *Confusion Matrix* adalah *accuracy*, *precision*, *recall*, dan *F1 score* yang digunakan sebagai penilaian performa dari sebuah model. *Accuracy* merupakan perhitungan pada banyaknya klasifikasi yang benar dari semua data.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.7)$$

Sementara itu, *precision* adalah perhitungan pada banyaknya klasifikasi data positif yang benar terhadap semua data yang diklasifikasikan positif.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.8)$$

Recall merupakan perhitungan pada banyaknya klasifikasi data positif yang benar diprediksi terhadap data positif yang aktual.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.9)$$

F1 Score merupakan nilai rata-rata diantara *precision* dan *recall*

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.10)$$

2.6 Penelitian Terdahulu

Ada sejumlah penelitian terkait mendeteksi indikasi depresi yang telah dilakukan. Penelitian Fine et al. (2020) melakukan analisis tingkat kecemasan, depresi, dan bunuh diri dari tanggal 1 Januari 2020 hingga 1 Juni 2020 dengan data yang dikumpulkan dari Twitter (X) di negara Amerika Serikat. Model yang digunakan adalah *Bi-directional Long Short-Term Memory* (BiLSTM) dengan GloVe sebagai *word embedding* untuk meminimalisir *overfitting*. Akurasi yang dihasilkan untuk memprediksi depresi dan bunuh diri adalah 72% dan 73%.

Selanjutnya, Penelitian yang dilakukan oleh Amanat et al. (2022) menggunakan *one-hot encoding* dan *principal component analysis* (PCA) sebagai teknik *pre-processing*. Penelitian ini menggunakan algoritma LSTM-RNN yang menggunakan 60 units dan hasil akurasi yang didapatkan berupa 0.98 *precision*, 0.99 *recall*, dan 0.98 *f1-measure*.

Kemudian, J. Singh et al. (2022) menggunakan *Long Short-Term Memory* (LSTM) untuk mendeteksi depresi dengan dataset yang berasal dari hasil *scraping* menggunakan *PushShift API* untuk mendapatkan postingan dari Reddit yang berhubungan dengan kesehatan mental dari beberapa *subreddit*. Penelitian ini melakukan *preprocessing* berupa *tokenization*, *data cleaning*, dan *stemming*. Lalu, data tersebut diubah menjadi vektor menggunakan GloVe *embedding* dan mendapatkan akurasi yaitu 94.88% dengan menggunakan algoritma LSTM dan GloVe *embedding*.

Selanjutnya, terdapat penelitian yang dilakukan oleh Jain et al. (2022) menggunakan model *Logistic Regression*, *Naïve Bayes*, *Support Vector Machines*, dan *Random Forest* untuk memprediksi apakah sebuah text berunsur depresi atau bunuh diri. Model dengan akurasi tertinggi adalah *Random Forest* dengan nilai akurasi sebesar 77.298%.

Lorenzoni et al. (2024) menggunakan *Random Forest Classifier*, *XGBoost*, dan *Support Vector Machine* untuk melakukan deteksi depresi yang bisa membedakan dengan PTSD (*Post-Traumatic Stress Disorder*). Dataset yang digunakan adalah *Distress Analysis Interview Corpus - Wizard of Oz* yang berisi data audio, video, dan transkrip dari interview yang dilakukan. Model *Random Forest Classifier* dan *XGBoost* menghasilkan akurasi tertinggi dengan nilai akurasi 83.8%.

Devika et al. (2023) menggunakan BERT sebagai *word embedding* dalam mengidentifikasi depresi dan bunuh diri menggunakan BERT-Based Approach. Penelitian ini melakukan *preprocessing* berupa *tokenization, removing stop words, dan part-of-speech tagging*. Lalu, data tersebut diubah menjadi vektor menggunakan BERT *embedding* dan menggunakan algoritma Bi-LSTM untuk melakukan prediksi. Dataset yang digunakan berasal dari media sosial Reddit. Bi-LSTM menghasilkan akurasi sebesar 81% dengan data yang sudah diubah menjadi vektor oleh BERT, sedangkan BERT model secara langsung hanya menghasilkan akurasi sebesar 72%.

Model BERT yang dikembangkan oleh Google banyak digunakan oleh peneliti-peneliti sehingga ada beberapa model yang dikembangkan dari BERT seperti ARABERT dan MARBERT yang digunakan oleh El-Ramly et al. (2021) dalam mendeteksi depresi. Dataset yang digunakan berasal dari CairoDep v1.0 yang memuat 7000 data depresi dan non-depresi dalam bahasa arab. Kemudian, dilakukan *data cleaning* dan *data preprocessing* sebelum digunakan untuk *training* pada model ARABERT dan MARBERT untuk melakukan prediksi. Akurasi yang diperoleh ARABERT adalah 96.93% sedangkan MARBERT menghasilkan akurasi yaitu 96.07%.

Hidayat & Maharani (2022) menggunakan IndoBERT dalam mendeteksi depresi yang berjudul dengan menggunakan data yang dikumpulkan dari Twitter (X). Selanjutnya, dilakukan *preprocessing data* agar data dapat digunakan untuk *training* pada IndoBERT model. Akurasi yang didapatkan masih pada angka 51%. Hal ini disebabkan oleh beberapa kata yang masih belum bisa diproses oleh IndoBERT seperti ‘wkwkwk’ dan kata-kata diluar Bahasa Indonesia.

Fadhel & Maharani (2024) melakukan deteksi depresi menggunakan model IndoBERTweet menggunakan data yang diperoleh dari Twitter (X). Langkah awal yang dilakukan dalam penelitian ini setelah melakukan pengumpulan dataset adalah melakukan *preprocessing* sampai pada tahap *tokenization*. Lalu, data teks yang telah menjadi token akan diproses oleh model IndoBERTweet untuk mendeteksi depresi. Akurasi yang dihasilkan berada pada angka 82% dan *F1-Score* sebesar 81%.

Penelitian Azahra & Setiawan (2023) membahas tentang *sentiment analysis* terhadap topik-topik yang ada pada media sosial. *Sentiment analysis* yang dilakukan menggunakan *granularity-oriented* yang lebih memfokuskan pada makna dari kata ataupun kalimat yang ada pada postingan di Twitter (X). *Preprocessing* dilakukan

sampai tahap *tokenization*, lalu dilakukan *labeling* secara manual. Setelah manual *labeling*, akan dilakukan *system labeling* dan dibandingkan hasilnya. Penelitian ini menggunakan algoritma LSTM dan IndoBERTweet method dalam melakukan klasifikasi. Akurasi tertinggi yang dihasilkan untuk *manual labeling* adalah 88.97%, sedangkan akurasi tertinggi untuk *system labeling* adalah 97.80%

Tabel 2.1 Penelitian Terdahulu

No	Penulis	Judul	Tahun	Keterangan
1.	Alex B. Fine, Patrick Crutchley, Jenny Blase, Joshua Carroll, dan Glen Coppersmith	Assessing population-level symptoms of anxiety, depression, and suicide risk in real time using NLP applied to social media data	2020	Penelitian ini melakukan analisis tingkat kecemasan, depresi, dan bunuh diri dari Twitter (X). Model yang digunakan adalah <i>Bi-directional Long Short-Term Memory (BiLSTM)</i> dengan <i>GloVe</i> dan memperoleh akurasi sebesar 72% dan 73% untuk memprediksi depresi dan bunuh diri.
2.	Amna Amanat,Muhammad Rizwan, Abdul Rehman Javed, Maha Abdelhaq, Raed Alsaqour ,Sharnil Pandya, dan Mueen Uddin	Deep Learning for Depression Detection from Textual Data	2022	Penelitian ini menggunakan algoritma LSTM-RNN yang menggunakan 60 <i>units</i> dan hasil akurasi yang didapatkan berupa 0.98 <i>precision</i> , 0.99 <i>recall</i> , dan 0.98 <i>f1-measure</i> . Tetapi, penelitian ini hanya menggunakan 4000 data text
3.	Jaskaran Singh, Mohammad Wzid, D.P. Singh, dan Sumit Pundir	An embedded LSTM based scheme for depression detection and analysis	2022	Penelitian ini melakukan data dari Reddit dan <i>preprocessing</i> berupa <i>tokenization</i> , <i>data cleaning</i> , dan <i>stemming</i> . Akurasi yang didapatkan dengan menggunakan algoritma LSTM dan <i>GloVe embedding</i> adalah 94.88%.

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

No	Penulis	Judul	Tahun	Keterangan
4.	Pratyaksh Jain , Karthik Ram Srinivas, dan Abhishek Vichare	Depression and Suicide Analysis Using Machine Learning and NLP	2022	Penelitian ini menggunakan model <i>Logistic Regression</i> , <i>Naïve Bayes</i> , <i>Support Vector Machines</i> , dan <i>Random Forest</i> untuk memprediksi apakah sebuah teks berunsur depresi atau bunuh diri dan menghasilkan nilai akurasi tertinggi yaitu 77.298% dengan model <i>Random Forest</i> .
5.	Guiliano Lorenzoni, Cristina Tavares, Nathalia Nascimento, Paulo Alencar, Donald Cowan David R. Cheriton	Assessing ML Classification Algorithms and NLP Techniques for Depression Detection: An Experimental Case Study	2024	Penelitian ini menggunakan dataset <i>Distress Analysis Interview Corpus - Wizardof-Oz</i> yang berisi data audio, video, dan transkrip interview dalam bahasa Mandarin. Akurasi paling tinggi dihasilkan oleh model <i>Random Forest Classifier</i> dan <i>XGBoost</i> dengan nilai akurasi 83.8%.
6.	S.P. Devika, M.R. Pooja, M.S. Arpitha, dan Ravi Vinayakumar	BERT-Based Approach for Suicide and Depression Identification	2023	Penelitian ini melakukan <i>preprocessing</i> berupa <i>tokenization</i> , <i>removing stop words</i> , dan <i>part-of-speech tagging</i> . Lalu, data diubah menjadi vektor menggunakan <i>BERT embedding</i> . Dataset yang digunakan berasal dari Reddit. Akurasi yang didapatkan ketika menggunakan <i>Bi-LSTM</i> adalah 81%, sedangkan <i>BERT</i> model menghasilkan akurasi sebesar 72%.

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

No	Penulis	Judul	Tahun	Keterangan
7.	Mohammad El-Ramly, Hager Abu-Elyazid, Youseef Mo'men, Gameel Alshaer, Nardine Adib, Kareen Alla Eldeen, dan Mariam El-Shazly	CairoDep: Detecting depression in Arabic posts using BERT transformers	2021	Penelitian ini menggunakan dataset CairoDep v1.0 yang memuat 7000 data depresi dan non-depresi dalam bahasa arab. Kemudian, dilakukan <i>data cleaning</i> dan <i>data preprocessing</i> sebelum digunakan untuk training pada model ARABERT dan MARBERT untuk melakukan prediksi. Akurasi yang diperoleh ARABERT adalah 96.93% sedangkan MARBERT menghasilkan akurasi yaitu 96.07%.
8.	Ilham Rizki Hidayat dan Warih Maharani	General Depression Detection Analysis Using IndoBERT Method	2022	Penelitian ini menggunakan data dari Twitter (X) dan dilakukan <i>preprocessing data</i> sebelum training pada IndoBERT model. Akurasi yang dihasilkan hanya 51% disebabkan oleh beberapa kata yang masih belum bisa diproses oleh IndoBERT seperti 'wkwkwk' dan kata-kata yang bukan Bahasa Indonesia. Penelitian ini mendeteksi depresi menggunakan model IndoBERTweet menggunakan data yang diperoleh dari Twitter (X). Data yang telah <i>preprocessing</i> akan diproses oleh model IndoBERTweet untuk mendeteksi depresi. Akurasi yang dihasilkan berada pada angka 82% dan <i>F1-Score</i> sebesar 81%.
9.	Muhammad Fadhel dan Warih Maharani	Depression Detection of Users in Social Media X using IndoBERTweet	2024	

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

No	Penulis	Judul	Tahun	Keterangan
10.	Nisa Maulia Azahra dan Erwin Budi Setiawan	Sentence-Level Granularity Oriented Sentiment Analysis of Social Media Using Long Short-Term Memory (LSTM) and IndoBERTweet Method	2023	Penelitian ini membahas tentang <i>sentiment analysis</i> terhadap topik-topik yang ada pada media sosial. Algoritma yang digunakan adalah LSTM dan IndoBERTweet. Akurasi tertinggi yang dihasilkan untuk <i>manual labeling</i> adalah 88.97%, sedangkan akurasi tertinggi untuk <i>system labeling</i> adalah 97.80%

2.7 Perbedaan Penelitian

Penelitian ini menggunakan algoritma LSTM dan IndoBERTweet sebagai *word embedding* dalam mendeteksi depresi. Perbedaan dari penelitian yang dilakukan oleh J. Singh et al. (2022) adalah menggunakan GloVe sebagai *word embedding* dan *data text* berbahasa Inggris. Lalu, Jain et al. (2022) dan Lorenzoni et al. (2024) menggunakan algoritma *Machine Learning* seperti *Random Forest Classifier* dan *Support Vector Machine* dalam melakukan deteksi depresi. Kemudian, perbedaan dari penelitian Devika et al. (2023) terdapat pada algoritma yang digunakan yaitu Bi-LSTM untuk mendeteksi depresi dan bunuh diri dan perbedaan pada Hidayat & Maharani (2022) adalah penelitian dilakukan menggunakan IndoBERT sebagai *word embedding* dan model dalam mendeteksi depresi. Penelitian Fadhel & Maharani (2024) hanya menggunakan IndoBERTweet sebagai model, sedangkan penelitian ini menggunakan IndoBERTweet sebagai *word embedding* dan digabungkan dengan algoritma LSTM untuk melakukan deteksi.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Data yang digunakan

Dataset yang digunakan dalam penelitian ini terdiri dari kumpulan postingan pada media sosial Twitter (X) dalam berbentuk teks yang telah dikumpulkan ke dalam file berekstensi .csv. Data yang dikumpulkan berjumlah 12703 data teks yang diperoleh dari Kaggle dan scraping secara manual. Dalam dataset terdapat kolom text yang menampung postingan Twitter (X) dan label untuk menampung tanda apakah postingan memiliki indikasi depresi atau tidak.

Postingan yang terindikasi depresi akan diberikan label 1 sedangkan postingan yang tidak terindikasi depresi akan diberikan label 0. Proses penlabelan dilakukan oleh penulis dan dipandu oleh seorang pakar psikologi bernama Natalia Nainggolan, S.Psi., M.Psi. Hal ini dilakukan agar data divalidasi sehingga data layak digunakan untuk tahap penelitian selanjutnya.

Tabel 3.1 Dataset Postingan Teks Normal dan Berindikasi Depresi

Text	Label
oh pantesan tadi pada rame, ternyata monek mau cb juni. cemas aku tuhâ˜º	0
Semakin bertambah usia, semakin cemas hidup.	0
kdg yang dibutuhin kalo lagi ada masalah adalah ketawa-ketawa sampe lupa kalo lagi ada masalah, pas inget lagi kayak gelisah aj gitu â€œeyahh anj ternyata msh ada beban sedihâ€¢asdfghjklzxccvnm	1
akhir" ini ngerasa takut & cemas banget mikirin hari h buat ospek kerja besuk seninđŸ~đŸ¥.jur takut banget karna aku bukan yg cpt banget deket sama temen,takut gabisa,takut ngecewain,dan kalo tiba tiba keinget langsung nangisđŸ~ž	1

Tabel 3.1 Dataset Postingan Teks Normal dan Berindikasi Depresi (Lanjutan)

Text	Label
Stres berkepanjangan,depresi setiap malam,merasa gelisah,udah bangkit berkali kali tapi hasilnya zonk, merasa mengandalkan Tuhan tidak ada gunanya ? Hei!! Cobalah untuk "menyerah" :)	1
Ini hari ga penting aku gaperlu cemas oke.	0
Cara nenangin diri karena cemas tuh caraku bikin seru sama asyik aja, dibayangin kalo hal itu emang se seru sama se asyik itu ^{~†~} ,	0
-dips! Cara hilangkn depresi gmn ya? Aku depresi bgt, gabutuh di smgtin.	1
Aku udh cape. Jangan nyuruh aku solat krna aku udh solat 5 wktu. Apakah ngerokok bakal nenangin pikiranku?	

Dataset kemudian dipecah menjadi *data training*, *data validation*, dan *data testing*. Tahap awal dataset dibagi adalah dengan membagi dataset menjadi *data training* dan *data testing* dengan proporsi 70:30, dimana 70% digunakan sebagai *data training* dan 30% sebagai *data testing*. Lalu, *data training* dipecah lagi dengan proporsi 50:20. 50% dialokasikan sebagai *data training* dan 20% sebagai *data validation*. Detail pemecahan dataset dicantumkan pada tabel 3.2

Tabel 3.2 Pemecahan Dataset

Data Training		Data Testing
Data Training Utama	Data Validation	
6351	2541	3811

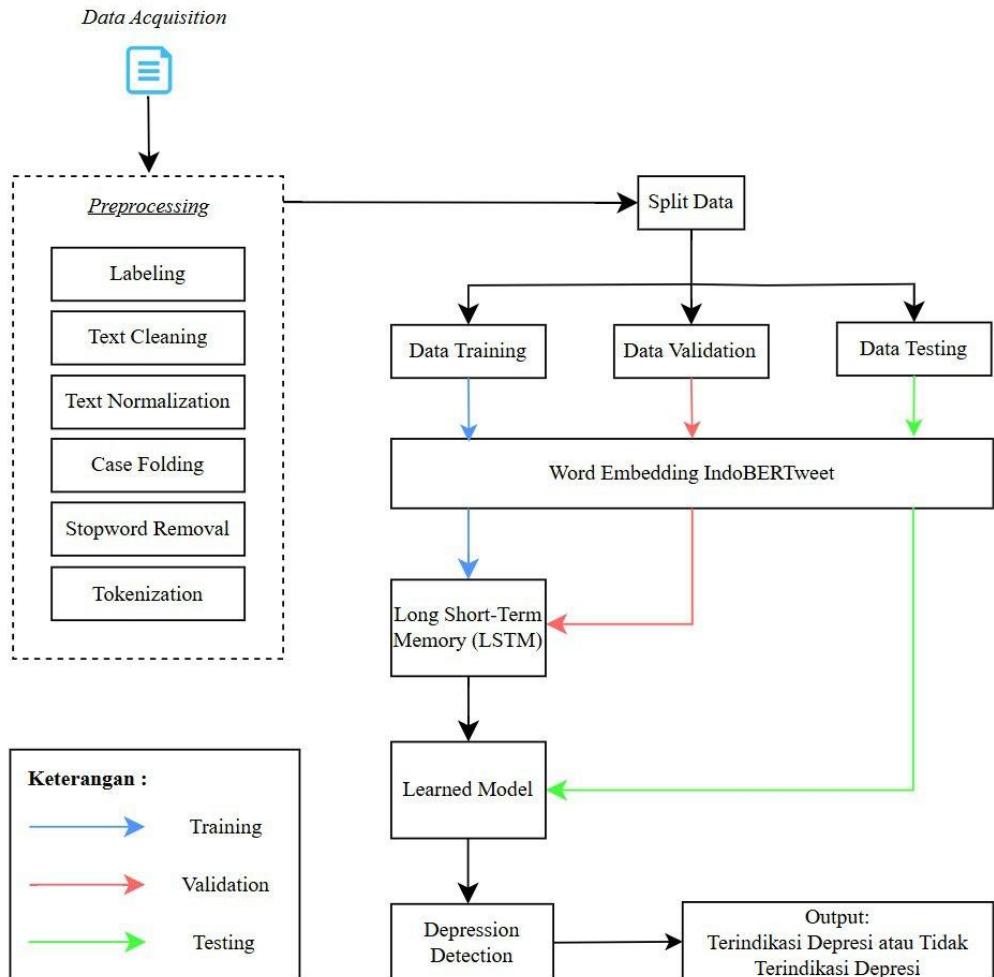
3.2 Arsitektur Umum

Penelitian ini dilakukan melalui beberapa tahapan. Tahap pertama adalah *data acquisition* atau pengumpulan data berupa data tweet yang akan dijadikan dataset dalam pengembangan model. Setelah dikumpulkan, akan dilakukan tahap *preprocessing* terhadap dataset yang terdiri dari *labeling*, *text cleaning*, *text normalization*, *case folding*, *stopword removal*, dan *tokenization*.

Proses *labeling* dilakukan terhadap setiap data tweet untuk menyatakan kelas objek berupa 1 untuk depresi dan 0 untuk tidak depresi. Setelah *labeling*, dilakukan *text cleaning* untuk membersihkan data dari karakter-karakter tertentu yang tidak diperlukan dalam proses deteksi seperti spasi yang berlebihan, link URL, dan lainnya agar dapat mengurangi tingkat kegagalan dalam proses mendeteksi. *Text normalization* adalah proses mengubah kata-kata slang, tidak baku, akronim, dan bahasa kolokial menjadi bahasa yang baku sesuai dengan KBBI. Selain itu, tahap ini juga memperbaiki kata-kata yang memiliki kesalahan ejaan agar dapat diubah ke kata yang benar.

Pada tahap *case folding*, dilakukan transformasi untuk setiap kata yang mengandung huruf besar menjadi huruf kecil agar token yang dihasilkan dalam *tokenization* tidak berlebih akibat perbedaan huruf. Contohnya: Sedih dan sedih akan memiliki token yang berbeda jika tidak diubah menjadi huruf kecil. Tahap *stopword removal* dilakukan untuk menghapus kata-kata yang tidak berpengaruh secara signifikan pada makna kalimat secara keseluruhan. Contoh dari kata-kata *stopword* aku, kamu, dia, dan, atau, tetapi, ke-, dari, dan lainnya. Tahap *tokenization* merupakan tahap terakhir dari *preprocessing* yang bertujuan untuk mengubah kalimat menjadi unit kecil yang disebut token. Untuk *word embedding* IndoBERTweet, diperlukan token khusus [CLS] pada awal kalimat dan [SEP] untuk akhir kalimat. Setelah tahap *tokenization*, akan dilanjutkan dengan tahap *word embedding* menggunakan IndoBERTweet.

Tahap selanjutnya adalah dataset dipisah menjadi 50% *data training*, 20% *validation*, dan 30% *testing* yang kemudian akan ditransformasi setiap data menjadi vektor oleh IndoBERTweet *Embedding*. Tahap terakhir adalah proses *training* menggunakan data yang telah diproses sebelumnya menggunakan algoritma *Long Short-Term Memory* (LSTM). Model yang dihasilkan akan diuji terhadap *data validation* sebelum dilakukan *testing*. Sistem yang dikembangkan akan menghasilkan output berupa teks mengandung depresi dan teks tidak mengandung depresi. Gambar 3.1 merupakan arsitektur umum dari proses penelitian yang dilakukan.



Gambar 3.1 Arsitektur Umum

3.2.1 Preprocessing

Preprocessing bertujuan mengolah data agar data menjadi lebih mudah diproses dan meningkatkan akurasi yang dihasilkan oleh model. Tahap-tahap yang terdapat pada *preprocessing* adalah *text cleaning*, *text normalization*, *case folding*, *stopword removal*, dan *tokenization*.

3.2.1.1 *Text Cleaning*

Text Cleaning merupakan tahap awal dalam *preprocessing* yang dilakukan untuk menghapus dan membersihkan teks dari elemen-elemen yang tidak digunakan oleh model untuk proses deteksi. Pada scenario pengujian pertama, elemen-elemen yang akan dihapus berupa emoji, emoticon, URL, simbol, tanda baca, angka, dan karakter-karakter non-ASCII. Proses *Text Cleaning* dapat dilihat dari pseudocode pada gambar 3.2 dan hasil dari implementasi *text cleaning* dapat dilihat pada tabel 3.3 dan 3.4.

FOR EACH row IN dataframe:

```

# Mengubah new lines menjadi spasi
row[“Preprocess”] ← REPLACE(row[“text”], “\n”, “ ”)

# Mengubah carriage return menjadi spasi
row[“Preprocess”] ← REPLACE(row[“Preprocess”], “\\r”, “ ”)

# Mengubah karakter ‘&’ menjadi kata ‘dan’
row[“Preprocess”] ← REPLACE(row[“Preprocess”], “&”, “dan”)

# Menghapus angka
row[“Preprocess”] ← REMOVE_NUMBERS(row[“Preprocess”])

# Menghapus URLs
row[“Preprocess”] ← REMOVE_URL(row[“Preprocess”])

# Menghapus emoji
row[“Preprocess”] ← REMOVE_EMOJI(row[“Preprocess”])

# Menghapus tanda baca
row[“Preprocess”] ← REMOVE_PUNCTUATION(row[“Preprocess”])

# Menghapus karakter non-ASCII
row[“Preprocess”] ← REMOVE_NON_ASCII(row[“Preprocess”])

# Menghapus karakter ‘@’
row[“Preprocess”] ← REPLACE(row[“Preprocess”], “@”, “ ”)

# Menghapus trailing dan leading spaces
row[“Preprocess”] ← TRIM(row[“Preprocess”])

# Menghapus extra spasi
row[“Preprocess”] ← REMOVE_EXTRA_SPACES(row[“Preprocess”])

```

END FOR

Gambar 3.2 Pseudocode *Text Cleaning*

Pada scenario pengujian kedua, penulis tidak melakukan penghapusan terhadap emoji, melainkan diubah menjadi teks yang mendeskripsikan emoji tersebut. Maka, tahapan menghapus emoji akan diubah menjadi `row[“Preprocess”] = EMOJI_TO_TEXT(row[“Preprocess”])`. Pengubahan ini memanfaatkan *library* yang tersedia yaitu *library* emoji dari Python.

Tabel 3.3 Implementasi *Text Cleaning* tanpa emoji

Sebelum Implementasi <i>Text Cleaning</i>	Setelah Implementasi <i>Text Cleaning</i>
Karena ayahku pernah blg œorang lain gaakan nyuruh kita utk bahagia jadi lebih baik kita membahagiakan diri sendiri←thatâ€™s why aku selalu kasih reward utk diriku dari setiap pencapaian yg udah aku lakuin hihiâ˜œi,↑œ https://t.co/E8h2qiYTDs	Karena ayahku pernah blg orang lain gaakan nyuruh kita utk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward utk diriku dari setiap pencapaian yg udah aku lakuin hihi

Tabel 3.4 Implementasi *Text Cleaning* dengan emoji

Sebelum Implementasi <i>Text Cleaning</i>	Setelah Implementasi <i>Text Cleaning</i>
Karena ayahku pernah blg œorang lain gaakan nyuruh kita utk bahagia jadi lebih baik kita membahagiakan diri sendiri←thatâ€™s why aku selalu kasih reward utk diriku dari setiap pencapaian yg udah aku lakuin hihiâ˜œi,↑œ https://t.co/E8h2qiYTDs	Karena ayahku pernah blg orang lain gaakan nyuruh kita utk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward utk diriku dari setiap pencapaian yg udah aku lakuin hihi berkilau

3.2.1.2 *Text Normalization*

Text Normalization merupakan tahap yang dilakukan untuk memperbaiki kata-kata yang tidak sesuai ejaan pada KBBI. Pseudocode pada gambar 3.3 merupakan proses dari *text normalization*.

```
FUNCTION normalizationText(text, corpus):
    # Membuat array baru untuk hasil normalisasi
    normalized ← []
    # Iterasi terhadap setiap kata pada teks
    FOR EACH word IN text:
        # Melakukan normalisasi kata
        normalizedText ← corpus.get(word, correctWord)
        # Menambahkan normalisasi kata pada array normalized
```

```

    normalized.append(normalizedText)
END FOR
RETURN “”.join(normalized)
END FUNCTION

```

Gambar 3.3 Pseudocode *Text Normalization*

Text normalization dilakukan dengan data yang diambil dari dataset indo4B yang berisikan kumpulan corpus berbahasa Indonesia. Hasil dari implementasi *Text Normalization* disajikan pada tabel 3.5.

Tabel 3.5 Implementasi *Text Normalization*

Sebelum Implementasi <i>Text Normalization</i>	Setelah Implementasi <i>Text Normalization</i>
Karena ayahku pernah blg orang lain gaakan nyuruh kita utk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward utk diriku dari setiap pencapaian yg udah aku lakuin hihi	Karena ayahku pernah bilang orang lain gaakan nyuruh kita untuk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward untuk diriku dari setiap pencapaian yang udah aku lakuin hihi

3.2.1.3 Case Folding

Case folding merupakan lanjutan pada *preprocessing* untuk mengubah semua huruf menjadi kecil. Tujuan dari *case folding* adalah agar semua karakter seragam dan mudah diproses oleh model. *Case folding* diterapkan dengan menggunakan *function String lower()* yang ditampilkan pada Tabel 3.6.

FUNCTION lower(inputString):

```

# Variabel untuk hasil case folding
result ← “”

# Iterasi setiap huruf dalam inputString
FOR EACH character IN inputString:
    # Melakukan pengecekan apakah huruf kapital atau huruf kecil
    IF character >= ‘A’ AND character <= ‘Z’:
        # Ubah ke huruf kecil dengan pengurangan kode ASCII

```

```

        result ← result + (character + ('a' – 'A'))
    ELSE:
        # Menambahkan huruf yang sudah kecil
        result ← result + character
    ENDIF
ENDFOR
RETURN result
END FUNCTION

```

Gambar 3.4 Pseudocode *Case Folding*Tabel 3.6 Implementasi *Case Folding*

Sebelum Implementasi <i>Case Folding</i>	Setelah Implementasi <i>Case Folding</i>
Karena ayahku pernah bilang orang lain gaakan nyuruh kita untuk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward untuk diriku dari setiap pencapaian yang udah aku lakuin hihi	karena ayahku pernah bilang orang lain gaakan nyuruh kita untuk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward untuk diriku dari setiap pencapaian yang udah aku lakuin hihi

3.2.1.4 *Stopword Removal*

Stopword Removal merupakan proses mengidentifikasi dan menghapus kata-kata yang kurang berkenaan dan yang tidak akan berpengaruh secara signifikan pada makna keseluruhan. Dalam implementasi proses ini, penulis menggunakan *library* Sastrawi yang menyajikan daftar serta contoh *stopword* berbahasa Indonesia. Selain itu, penulis juga menambah beberapa *stopword* seperti adalah, kan, dan dong. Pseudocode dari proses *Stopword Removal* disajikan pada gambar 3.5 serta implementasi hasilnya pada Tabel 3.7.

```

stop_factory ← StopWordRemoverFactory().get_stop_words()
more_stopword ← ['adalah', 'kan', 'dong']
data ← stop_factory + more_stopword
swr ← StopWordRemover(data)
result ← []

```

```
FOR EACH word IN row[“Preprocess”]:
```

```
    # Iterasi terhadap setiap stopword pada swr
```

```
    FOR EACH stopword IN swr:
```

```
        IF word IS NOT EQUAL TO stopword:
```

```
            # Menghapus stopword
```

```
            result.append(word)
```

```
    END FOR
```

```
END FOR
```

```
RETURN “”.join(result)
```

Gambar 3.5 Pseudocode *Stopword Removal*

Tabel 3.7 Implementasi *Stopword Removal*

Sebelum Implementasi <i>Stopword Removal</i>	Setelah Implementasi <i>Stopword Removal</i>
karena ayahku pernah bilang orang lain gaakan nyuruh kita untuk bahagia jadi lebih baik kita membahagiakan diri sendiri thats why aku selalu kasih reward untuk diriku dari setiap pencapaian yang udah aku lakuin hihi	karena ayahku pernah bilang orang lain gaakan nyuruh bahagia jadi lebih baik membahagiakan diri sendiri thats why aku selalu kasih reward diriku dari setiap pencapaian udah aku lakuin hihi

3.2.1.5 Tokenization

Tokenization merupakan proses yang penting untuk membagi teks berupa kalimat menjadi unit-unit tunggal yang disebut dengan *token*. *Tokenization* dilakukan dengan memanfaatkan indobertweet-base-uncased yang menyediakan *tokenizer* dan model yang telah di pre-train menggunakan korpus bahasa Indonesia. Model BERT memerlukan token khusus pada setiap awal dan akhir kalimat dengan token [CLS] sebagai *Classification token* dan [SEP] sebagai *Separator token*. Setelah itu, *tokenizer* akan memeriksa setiap kata pada suatu kalimat untuk memberikan token. Jika terdapat kata yang tidak ada dalam *vocabulary tokenizer*, maka *tokenizer* akan mencoba memisahkan suatu kata menjadi beberapa bagian agar sesuai dengan *vocabulary* yang ada. Kata yang tidak dikenali akan ditandai dengan simbol ###. Gambar 3.6 adalah pseudocode proses *tokenization* menggunakan *tokenizer* IndoBERTweet.

```

FUNCTION tokenizeText(text):
    # Membuat array token
    tokens ← []
    # Iterasi mengubah setiap kata menjadi token
    FOR EACH word IN text:
        # Mengubah kata menjadi token
        token ← word
        # Mengubah token menjadi id token
        token_id ← token
        tokens.append(token_id)
    END FOR
    RETURN tokens
END FUNCTION

```

Gambar 3.6 Pseudocode *Tokenization*

Token yang telah dibuat akan diubah menjadi *data numeric* yang merepresentasikan *token*. Berikut hasil dari *tokenization* pada tabel 3.8 dan perubahan *token* menjadi *data numeric* pada tabel 3.9

Tabel 3.8 Implementasi *Tokenization*

Sebelum Implementasi <i>Tokenization</i>	Setelah Implementasi <i>Tokenization</i>
karena ayahku pernah bilang orang lain gaakan nyuruh bahagia jadi lebih baik membahagiakan diri sendiri thats why aku selalu kasih reward diriku dari setiap pencapaian udah aku lakuin hihi	'[CLS]', 'karena', 'ayahku', 'pernah', 'bilang', 'orang', 'lain', 'gaakan', 'nyuruh', 'bahagia', 'jadi', 'lebih', 'baik', 'membahagiakan', 'diri', 'sendiri', 'thats', 'why', 'aku', 'selalu', 'kasih', 'reward', 'diriku', 'dari', 'setiap', 'pencapaian', 'udah', 'aku', 'lakuin', 'hihi', '[SEP]'

Tabel 3.9 Pengubahan Token menjadi ID

Sebelum menjadi ID	Sesudah menjadi ID
'[CLS]', 'karena', 'ayahku', 'pernah', 'bilang', 'orang', 'lain', 'gaakan', 'nyuruh', 'bahagia', 'jadi', 'lebih', 'baik', 'membahagiakan', 'diri', 'sendiri', 'thats', 'why', 'aku', 'selalu', 'kasih', 'reward', 'diriku', 'dari', 'setiap', 'pencapaian', 'udah', 'aku', 'lakuin', 'hihi', '[SEP]'	[3, 1686, 25689, 2137, 7141, 1646, 1690, 13576, 10172, 8048, 2359, 1716, 1983, 26100, 2129, 1979, 30600, 21226, 2254, 2643, 3774, 14412, 13314, 1542, 2189, 9361, 9988, 2254, 13789, 12521, 4]

3.2.2 *Split Dataset*

Dataset yang telah dilakukan *preprocessing* akan dipecah menjadi *data training*, *data validation*, dan *data testing*. Dataset dibagi menggunakan rasio 50:20:30, dimana 50% sebagai *data training*, 20% sebagai *data validation*, dan 30% sebagai *data testing*. *Data training* dan *data validation* digunakan dalam proses membuat dan melatih model. Kemudian, kemampuan model akan diuji menggunakan *data testing* dalam mendeteksi indikasi depresi.

3.2.3 *Word Embedding*

Pada tahap *Word Embedding*, data teks akan diubah menjadi data numerik berbentuk vektor sebagai representasi kata. *Word Embedding* yang digunakan untuk penelitian ini merupakan model *indobertweet-base-uncased* dari *huggingface transformer*.

Model IndoBERTweet merupakan turunan dari model BERT dan IndoBERT yang dilakukan pre-training menggunakan data twitter berbahasa Indonesia. IndoBERTweet sebagai *word embedding* memanfaatkan arsitektur *Transformer* untuk membentuk representasi kata berbasis konteks. Untuk menghasilkan representasi kata, IndoBERTweet menggunakan *self-attention* pada 12 lapisan *encoder Transformer* yang memungkinkan untuk memperhatikan hubungan setiap token sehingga representasi vektor yang dihasilkan menjadi kontekstual dan mencerminkan makna kata dalam kalimat tertentu. Model ini menggunakan pendekatan *WordPiece tokenization* untuk memecah kata menjadi sub-kata agar kata-kata baru atau langka tetap dapat diwakili dalam bentuk vektor (Koto et al., 2021).

Embedding yang dihasilkan oleh IndoBERTweet memiliki dimensi 768 sebanyak 12 vektor. Untuk penelitian ini, hanya vektor yang dihasilkan oleh *last hidden state* akan digunakan untuk diproses oleh model. Contoh vektor yang dihasilkan IndoBERTweet sebagai *word embedding* pada kalimat ‘karena ayahku pernah bilang orang lain gaakan nyuruh bahagia jadi lebih baik membahagiakan diri sendiri thats why aku selalu kasih reward diriku dari setiap pencapaian udah aku lakuin hihi’ disajikan pada gambar 3.2

```
tensor([[[[-0.0317, -0.2656, -0.5966, ..., -1.0354, -1.0820, 1.3519],
          [-0.1154, 0.4819, -0.1212, ..., 0.2528, -0.8534, 2.1462],
          [-1.1443, -0.2096, 0.0864, ..., 0.8911, 0.1116, 1.5740],
          ...,
          [-0.3423, 0.8669, 0.6571, ..., -1.5514, -0.0455, 0.6971],
          [-1.2026, -0.7762, 0.0225, ..., 0.0872, 0.4624, 1.3652],
          [-0.6452, -0.7529, -0.0987, ..., 0.3357, 0.0854, 2.3148]]])
```

Gambar 3.7 Hasil *Word Embedding*

3.2.4 Long Short-Term Memory (*LSTM*)

Pengembangan model dilakukan dengan algoritma *Long Short-Term Memory (LSTM)*. Algoritma ini mampu menampung teks yang panjang dan menangkap hubungan temporal antar kata. Proses pelatihan model akan dijelaskan secara terperinci sebagai berikut.

1. *Data training* dan *validation* yang telah diubah menjadi vektor menggunakan model IndoBERTweet sebagai *word embedding* akan digunakan sebagai input dalam membangun model
2. Tahap selanjutnya adalah menentukan jumlah lapisan (*layers*), jumlah *units* pada setiap lapisan, dan fungsi aktivasi yang digunakan pada *layers*.
3. Model LSTM menerima input dari *data training* dalam bentuk urutan (*sequence*) dan mengolahnya dengan mempertimbangkan informasi dari waktu sebelumnya. Proses ini melibatkan propagasi maju (*forward propagation*) untuk menghasilkan output dan propagasi mundur (*backward propagation*) untuk menghitung gradien dan memperbarui bobot.
4. Data input yang diterima akan digunakan untuk menghitung *cell state* dan *output gate* sebagai komponen dari *last hidden state* LSTM. *Last hidden state* dari LSTM menampung informasi yang telah diproses dan dikumpulkan oleh LSTM sebagai output final.

5. Hasil *training* dan *validation* dihitung berdasarkan *accuracy* dan *loss function* demi mengukur seberapa baik model yang dibentuk dalam melakukan prediksi. *Loss function* yang digunakan adalah *binary cross-entropy function* yang dikhususkan dalam memprediksi label *binary*.
6. Model yang telah terbentuk dan menghasilkan *accuracy* yang bagus akan dilakukan uji terhadap *data testing* dan dievaluasi kinerjanya menggunakan *confusion matrix* untuk menghitung *accuracy*, *precision*, *recall*, dan *f-1 score*.

IndoBERTweet *embedding* menghasilkan vektor dengan dimensi 768 sehingga untuk memudahkan penjelasan perhitungan, vektor yang digunakan akan berdimensi 3 sebagai representasi dari vektor *embedding* yang telah dihasilkan. Berikut adalah rincian perhitungan *last hidden state* dari algoritma LSTM dari kalimat “aku memilih terus percaya bahwa apa terjadi adalah sebaik baiknya rencana nya“ yang telah diubah menjadi *word embedding*.

1. Menetapkan variabel x_t , h_{t-1} , dan C_{t-1}

x_t merupakan vektor input yang berdimensi 768. Vektor ini akan kita sederhanakan menjadi dimensi 3 agar mempermudah penjelasan perhitungan pada subbab ini. h_{t-1} dan C_{t-1} merupakan *hidden state* dan *cell state* dari output LSTM sebelumnya. Jika tidak terdapat output sebelumnya, maka *hidden state* dan *cell state* sebelumnya akan bernilai 0.

$$x_t = [0.5, 0.2, 0.1]$$

$$h_{t-1} = [0.1, 0.4, 0.2]$$

$$C_{t-1} = [0.6, 0.2, 0.4]$$

2. Menetapkan Bobot dan Bias

Bobot dan bias diperlukan agar data dapat ditransformasi dengan baik dan memberikan kemampuan kepada LSTM untuk mengelola data yang kompleks dan rumit. Bobot dan bias akan ditetapkan pada setiap *gate* dan *cell state*.

Perhitungan Bobot dan bias untuk *input gate*:

$$W_i = \begin{bmatrix} 0.2 & 0.5 & 0.1 \\ 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$

$$b_i = [0.1, 0.2, 0.3]$$

Perhitungan Bobot dan bias untuk *forget gate*:

$$W_f = \begin{bmatrix} 0.4 & 0.3 & 0.1 \\ 0.2 & 0.4 & 0.3 \\ 0.5 & 0.6 & 0.1 \end{bmatrix}$$

$$b_f = [0.3, 0.2, 0.1]$$

Perhitungan Bobot dan bias untuk *cell state*:

$$W_c = \begin{bmatrix} 0.3 & 0.7 & 0.2 \\ 0.4 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.8 \end{bmatrix}$$

$$b_c = [0.2, 0.1, 0.3]$$

Perhitungan Bobot dan bias untuk *output gate*:

$$W_o = \begin{bmatrix} 0.2 & 0.1 & 0.3 \\ 0.5 & 0.4 & 0.2 \\ 0.6 & 0.2 & 0.1 \end{bmatrix}$$

$$b_o = [0.1, 0.2, 0.1]$$

3. Menghitung *Input Gate* (i_t)

Langkah pertama adalah menghitung *input gate* (i_t) yang bertujuan untuk menentukan seberapa banyak informasi yang akan dipertimbangkan dan dipilih untuk memasuki *cell state*. Dalam perhitungan ini, bobot dikalikan untuk input gate dengan kombinasi dari *hidden state* sebelumnya dan input saat ini, lalu menambahkan bias yang telah ditetapkan. Setelah itu, fungsi aktivasi sigmoid diterapkan guna memperoleh nilai antara 0 dan 1 yang mewakili proporsi informasi yang akan dimasukan atau tidak pada *cell state*.

Proses Perhitungan *Input Gate* menggunakan persamaan 2.2:

$$[h_{t-1}, x_t] = [0.1, 0.4, 0.2, 0.5, 0.2, 0.1]$$

$$W_i \cdot [h_{t-1}, x_t] = \begin{bmatrix} 0.2 & 0.5 & 0.1 \\ 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 0.1 \\ 0.4 \\ 0.2 \\ 0.5 \\ 0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.44 \\ 0.51 \\ 0.44 \end{bmatrix}$$

$$W_i \cdot [h_{t-1}, x_t] + b_i = \begin{bmatrix} 0.44 \\ 0.51 \\ 0.44 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.54 \\ 0.71 \\ 0.74 \end{bmatrix}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) = \sigma([0.54, 0.71, 0.74]) \approx [0.635, 0.670, 0.676]$$

4. Menghitung *forget gate* (f_t)

Langkah selanjutnya adalah menghitung *forget gate* (f_t) yang bertujuan menentukan informasi yang akan dihapus dari *cell state* sebelumnya. Proses perhitungan *forget gate* mirip dengan input gate, di mana bobot untuk *forget gate* dikalikan dengan kombinasi dari *hidden state* sebelumnya dan input saat ini, dan kemudian bias ditambahkan. Hasilnya juga diterapkan fungsi aktivasi sigmoid untuk memperoleh nilai antara 0 dan 1 yang mewakili seberapa banyak informasi yang akan dilupakan dari *cell state*. Proses Perhitungan *Forget Gate* menggunakan persamaan 2.1:

$$[h_{t-1}, x_t] = [0.1, 0.4, 0.2, 0.5, 0.2, 0.1]$$

$$W_f \cdot [h_{t-1}, x_t] = \begin{bmatrix} 0.4 & 0.3 & 0.1 \\ 0.2 & 0.4 & 0.3 \\ 0.5 & 0.6 & 0.1 \end{bmatrix} \cdot \begin{bmatrix} 0.1 \\ 0.4 \\ 0.2 \\ 0.5 \\ 0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.39 \\ 0.39 \\ 0.71 \end{bmatrix}$$

$$W_f \cdot [h_{t-1}, x_t] + b_f = \begin{bmatrix} 0.39 \\ 0.39 \\ 0.71 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.69 \\ 0.59 \\ 0.81 \end{bmatrix}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) = \sigma([0.69, 0.59, 0.81]) \approx [0.666, 0.645, 0.693]$$

5. Menghitung *Candidate Cell State*

Setelah menghitung *forget gate*, langkah selanjutnya adalah menghitung *candidate cell state* (\tilde{C}_t) dengan mengalikan bobot untuk *cell state* dengan kombinasi dari *hidden state* sebelumnya dan input saat ini, ditambah bias. Kemudian, fungsi tanh diterapkan untuk menghasilkan nilai-nilai yang akan digunakan untuk memperbarui *cell state*. *Candidate cell state* mewakili informasi baru yang akan ditambahkan ke dalam *cell state* baru. Proses Perhitungan *Candidate Cell Gate* menggunakan persamaan 2.3:

$$[h_{t-1}, x_t] = [0.1, 0.4, 0.2, 0.5, 0.2, 0.1]$$

$$W_C \cdot [h_{t-1}, x_t] = \begin{bmatrix} 0.3 & 0.7 & 0.2 \\ 0.4 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.8 \end{bmatrix} \cdot \begin{bmatrix} 0.1 \\ 0.4 \\ 0.2 \\ 0.5 \\ 0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.68 \\ 0.63 \\ 0.43 \end{bmatrix}$$

$$W_C \cdot [h_{t-1}, x_t] + b_C = \begin{bmatrix} 0.68 \\ 0.63 \\ 0.43 \end{bmatrix} + \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.88 \\ 0.73 \\ 0.73 \end{bmatrix}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$= \tanh([0.88, 0.73, 0.73]) \approx [0.707, 0.634, 0.634]$$

6. Menghitung *Cell State* (C_t)

Setelah mendapatkan *input gate*, *forget gate*, dan *candidate cell state*, tahap berikutnya adalah menghitung *cell state baru* (C_t) dengan menggabungkan informasi dari *cell state* sebelumnya dan *candidate cell state*, dikalikan dengan nilai dari *forget gate* dan *input gate*. Perhitungan *Cell State* dilakukan untuk memastikan bahwa informasi yang tidak relevan dihapus sementara informasi yang baru ditambahkan. Proses Perhitungan *Cell Gate* menggunakan persamaan 2.4:

$$\begin{aligned} C_t &= [0.666, 0.645, 0.693] \cdot [0.6, 0.2, 0.4] \\ &\quad + [0.635, 0.680, 0.676] \cdot [0.707, 0.634, 0.634] \\ C_t &= \begin{bmatrix} 0.3996 + 0.129 + 0.2772 \\ 0.449245 + 0.42418 + 0.429384 \end{bmatrix} = \begin{bmatrix} 0.8058 \\ 1.301809 \end{bmatrix} \end{aligned}$$

7. Menghitung *Output Gate* (o_t)

Output gate (o_t) mewakili seberapa banyak informasi dari *cell state* yang akan digunakan untuk menghasilkan *hidden state*. Dengan cara yang sama seperti langkah sebelumnya, kita mengalikan bobot untuk *output gate* dengan kombinasi dari *hidden state* sebelumnya dan input saat ini, menambahkan bias, dan menerapkan fungsi aktivasi sigmoid. Proses Perhitungan *Output Gate* menggunakan persamaan 2.5:

$$[h_{t-1}, x_t] = [0.1, 0.4, 0.2, 0.5, 0.2, 0.1]$$

$$W_o \cdot [h_{t-1}, x_t] = \begin{bmatrix} 0.2 & 0.1 & 0.3 \\ 0.5 & 0.4 & 0.2 \\ 0. & 0.2 & 0.1 \end{bmatrix} \cdot \begin{bmatrix} 0.1 \\ 0.4 \\ 0.2 \\ 0.5 \\ 0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.47 \\ 0.63 \\ 0.46 \end{bmatrix}$$

$$W_o \cdot [h_{t-1}, x_t] + b_o = \begin{bmatrix} 0.47 \\ 0.63 \\ 0.46 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.57 \\ 0.83 \\ 0.56 \end{bmatrix}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) = \sigma([0.57, 0.83, 0.56]) \approx [0.639, 0.695, 0.636]$$

8. Menghitung *Last Hidden State*

Tahap terakhir adalah *last hidden state* (h_t) yang merupakan output utama dari LSTM. *Last hidden state* dihasilkan dengan mengalikan *output gate* dengan fungsi tanh dari *cell state* saat ini. *Hidden state* ini menyimpan informasi yang relevan dari input yang telah diproses dan akan digunakan pada langkah-langkah berikutnya. Proses Perhitungan *Last Hidden State* menggunakan persamaan 2.6:

$$h_t = o_t * \tanh(C_t)$$

$$h_t = [0.639, 0.695, 0.636] \cdot \tanh([0.8058, 1.301809])$$

$$\tanh([0.8058, 1.301809]) \approx [0.665, 0.860]$$

$$h_t = [0.639, 0.695, 0.636] \cdot [0.665, 0.860] \approx [0.425, 0.598, 0.546]$$

3.2.5 Output

Model yang telah dihasilkan dapat digunakan untuk mendeteksi apakah sebuah teks memiliki indikasi depresi atau tidak.

3.3 Perancangan Sistem

Pada bagian ini, dijelaskan rincian rancangan sistem antarmuka yang dibentuk untuk mendeteksi indikasi depresi pada teks postingan sosial media. Sistem yang dikembangkan berbasis website dan *google chrome extension*. Penjelasan yang akan diuraikan bertujuan untuk memberikan pemahaman secara mendalam sekaligus menjadi panduan dalam pengembangan sistem.

3.3.1 Desain Halaman *Training*

Halaman *Training* dibuat untuk menyediakan tempat bagi pengguna melakukan pelatihan model menggunakan file dataset yang berformat .csv atau .xlsx. File yang berhasil diupload akan digunakan untuk proses *training* pada model. Setelah selesai, halaman *training* akan menampilkan tabel yang menampung data sebelum dan sesudah hasil *preprocessing* sebagai sarana pembantu bagi pengguna. Desain halaman *training* dipaparkan pada gambar 3.8.

Hasil Preprocessing	
Text	Preprocessed Text

Gambar 3.8 Desain Antarmuka Halaman *Training*

Keterangan:

Label 1: Tempat untuk pengguna mengupload file dataset berformat .csv atau .xlsx

Label 2: Tombol untuk mengupload file ke model dan inisiasi *training*

Label 3: Tabel yang menampung data sebelum dan sesudah dilakukan *preprocessing*

3.3.2 Desain Halaman *Testing*

Halaman *Testing* merupakan tempat bagi pengguna untuk melakukan evaluasi terhadap model yang telah ditraining dengan menggunakan file dataset yang berformat .csv atau .xlsx. File yang telah berhasil diupload akan dilakukan proses evaluasi model. Setelah selesai, halaman *testing* akan menampilkan tabel yang menampung data sebelum dan sesudah hasil *preprocessing* beserta label asli dan label

yang diprediksi menggunakan model LSTM. Selain itu, halaman *testing* juga menampilkan hasil evaluasi menggunakan *confusion matrix* yang mencakup nilai *accuracy*, *precision*, *recall*, dan *f-1 score*. Desain halaman testing disajikan pada gambar 3.9.

SenseDepresi

Home Training Testing Cek Depresi Teks Cek Depresi File

Testing

1

2

3

Text	Preprocessed Text	Label	Label Predict

Gambar 3.9 Desain Antarmuka Halaman Testing

Keterangan:

Label 1: Tempat untuk pengguna mengupload file dataset berformat .csv atau .xlsx

Label 2: Tombol untuk mengupload file ke model dan inisiasi *testing*

Label 3: Tabel yang menampung data sebelum dan sesudah dilakukan *preprocessing* serta label asli dan label yang diprediksi oleh model

3.3.3 Desain Halaman Cek Depresi Teks

Halaman Cek Depresi Teks disediakan agar pengguna dapat mendeteksi apakah sebuah postingan teks memiliki indikasi depresi atau tidak dari input yang diberikan. Pengguna memasukan postingan teks yang ingin diperiksa pada *textbox*, lalu menekan tombol “Cek Pesan” agar ditampilkan hasil dari deteksi antara terindikasi depresi atau tidak terindikasi depresi. Desain Halaman Cek Depresi Teks dipaparkan pada gambar 3.10

SenseDepresi

Home Training Testing Cek Depresi Teks Cek Depresi File

Selamat Datang di SenseDepresi

Deteksi text postingan yang mungkin memiliki indikasi depresi

Masukkan Pesan:

1

2

3

Gambar 3.10 Desain Antarmuka Halaman Cek Indikasi Depresi Teks

Keterangan:

Label 1: Textbox yang menampung input postingan teks pengguna yang akan dideteksi

Label 2: Tombol untuk memulai proses deteksi terhadap input yang telah diberikan

Label 3: Label yang menampung hasil prediksi yang telah dilakukan antara terindikasi depresi dan tidak terindikasi depresi

3.3.4 Desain Halaman Cek Depresi File

Halaman Cek Depresi File disediakan agar pengguna dapat mendeteksi postingan teks menggunakan file berformat .csv atau .xlsx. Pengguna mengupload file yang ingin dideteksi, lalu menekan tombol upload dan cek file untuk memulai proses deteksi terhadap data yang ada pada file yang telah diupload. Hasil deteksi yang telah dilakukan akan ditampilkan dalam bentuk tabel yang memuat data teks dan label hasil deteksi oleh model LSTM. Desain Halaman Cek Depresi File disajikan pada gambar 3.11 dibawah ini.

The screenshot shows a user interface for depression detection. At the top left is the logo 'SenseDepresi'. Along the top right are navigation links: Home, Training, Testing, Cek Depresi Teks, and Cek Depresi File. Below this is a main section titled 'Upload File untuk Deteksi Depresi'. It contains four numbered labels: 1 points to a white input field labeled 'Input File dataset'; 2 points to a grey button labeled 'Upload dan Cek'; 3 points to a table with two columns, 'Text' and 'Label', containing five rows of data; and 4 points to a grey button labeled 'Download Hasil'.

Gambar 3.11 Desain Antarmuka Halaman Cek Indikasi Depresi File

Keterangan:

Label 1: Tempat untuk pengguna mengupload file yang ingin dilakukan deteksi berformat .csv atau .xlsx

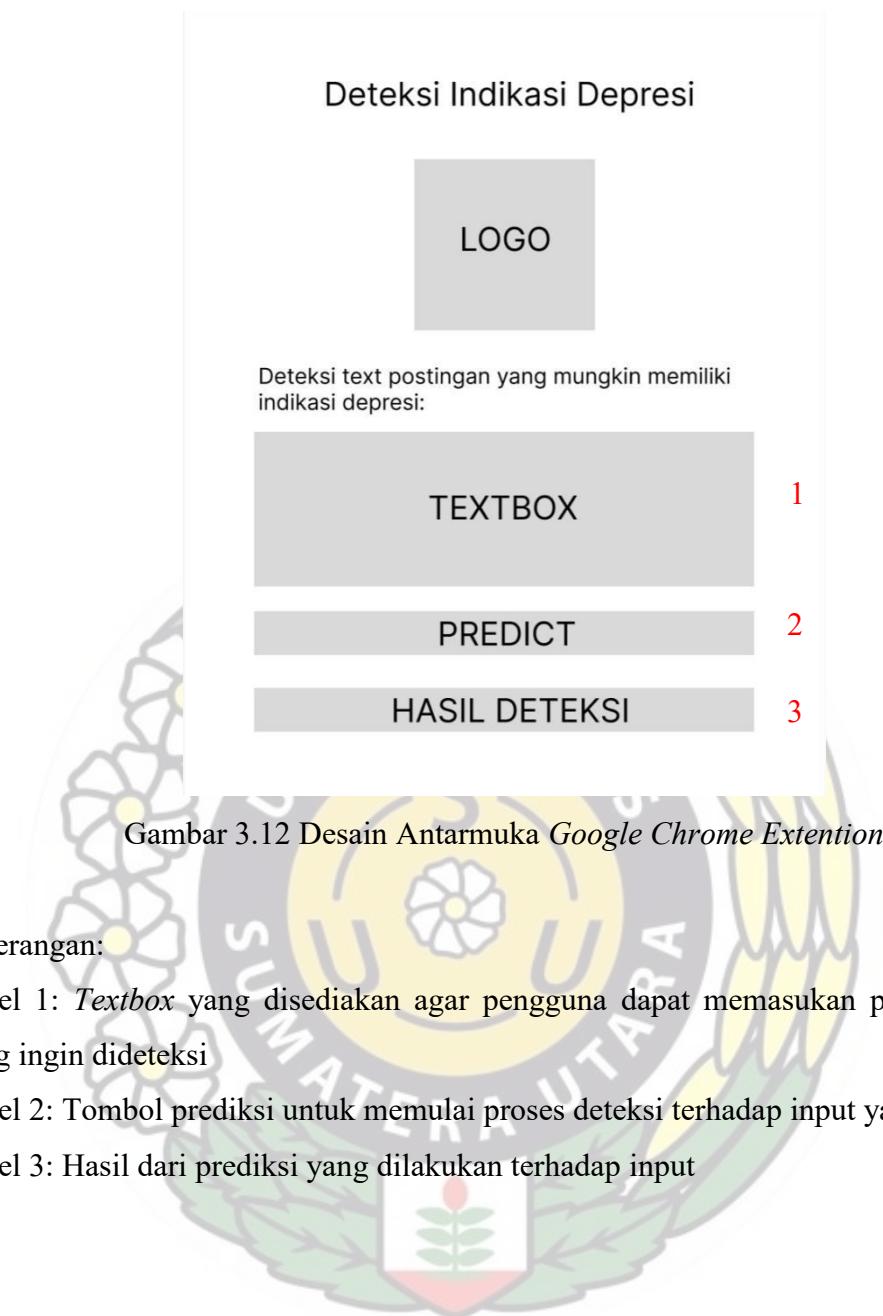
Label 2: Tombol untuk mengupload file ke model dan memulai proses deteksi

Label 3: Tabel yang memuat data teks dan label hasil prediksi dari model

Label 4: Tombol untuk mendownload file yang menampung data teks dan label hasil prediksi

3.3.5 Desain *Chrome Extension*

Chrome extension dibuat agar pengguna dapat mengakses sistem secara langsung tanpa harus masuk kedalam website. Hal ini bertujuan untuk memudahkan pengguna dalam mengakses sistem mendeteksi indikasi depresi. *Chrome extension* yang dibuat hanya bisa melakukan prediksi terhadap input text yang diberikan. Pengguna memasukan postingan teks dalam textbox yang disediakan, lalu menekan tombol prediksi untuk mendapatkan hasil prediksi antara terindikasi depresi atau tidak terindikasi depresi. Desain *Chrome extension* dipaparkan pada gambar 3.12.



BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Pengembangan sistem mendetksi indikasi depresi postingan teks pada media sosial dilakukan melalui implementasi IndoBERTweet sebagai *word embedding* dan algoritma *Long Short-Term Memory (LSTM)*. Proses implementasi sistem melibatkan penggunaan perangkat keras dan lunak, yaitu:

4.1.1 Spesifikasi Perangkat Keras dan Lunak

Berikut adalah spesifikasi perangkat keras yang dipakai untuk mengembangkan sistem dalam penelitian ini:

1. Laptop Lenovo Legion Pro 5 16IRX8
2. Prosesor 13th Gen Intel(R) Core(TM) i9-13900HX @ 2.20 GHz
3. Daya tampung memori 32 GB

Berikut merupakan spesifikasi perangkat lunak yang dipakai untuk mengembangkan sistem dalam penelitian ini:

1. OS: Windows 11 Home
2. Bahasa Pemrograman Python versi 3.12.2
3. IDE: *Microsoft Visual Studio Code*

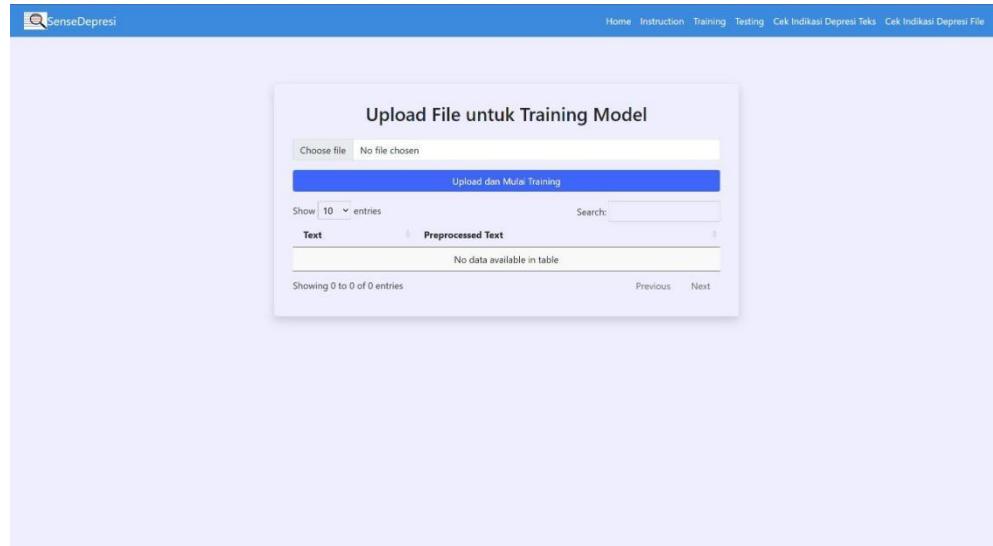
4.1.2 Penerapan sistem berbasis Website dan *Google Chrome extension*

Pada tahap ini, akan dipaparkan hasil implementasi dari rancangan sistem yang telah dibentuk sebelumnya. Sistem ini memiliki empat menu utama yaitu halaman *training*, halaman *testing*, halaman cek depresi teks, dan halaman cek depresi file.

1. Halaman Training

Halaman Training dapat digunakan oleh pengguna untuk melakukan proses pelatihan model. Pengguna dapat mengupload file *.csv* atau *.xlsx* sebagai file dataset, lalu menekan tombol yang tersedia untuk memulai proses *training*.

model. Setelah selesai, halaman akan menampilkan tabel yang memuat data teks sebelum dan sesudah preprocessing. Tampilan halaman training disajikan pada gambar 4.1 dan 4.2



Gambar 4.1 Halaman *Training*

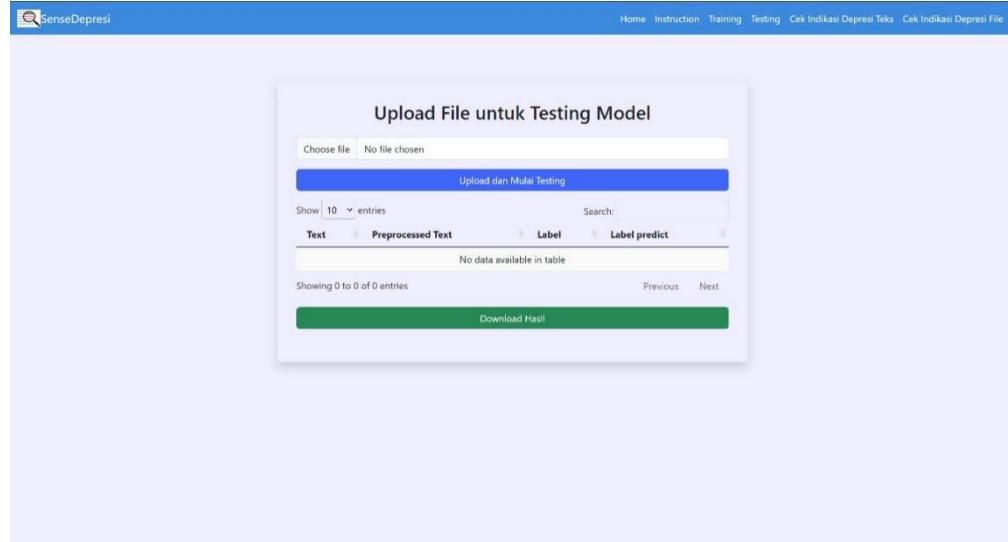
Upload File untuk Training Model	
Choose file	train_data.xlsx
Upload dan Mulai Training	
Show 10 entries	Search:
Text	Preprocessed Text
No data available in table	
Showing 0 to 0 of 0 entries	
Previous	Next

Gambar 4.2 Hasil *Training*

2. Halaman Testing

Halaman Testing disediakan untuk melakukan evaluasi terhadap model yang telah dilatih sebelumnya. Pengguna dapat mengupload file *.csv* atau *.xlsx* sebagai file dataset, lalu menekan tombol yang tersedia untuk memulai proses evaluasi model. Setelah selesai halaman akan menampilkan tabel yang memuat data teks sebelum dan sesudah preprocessing beserta label asli dan label hasil prediksi model. Selain itu, halaman testing juga menampilkan evaluasi

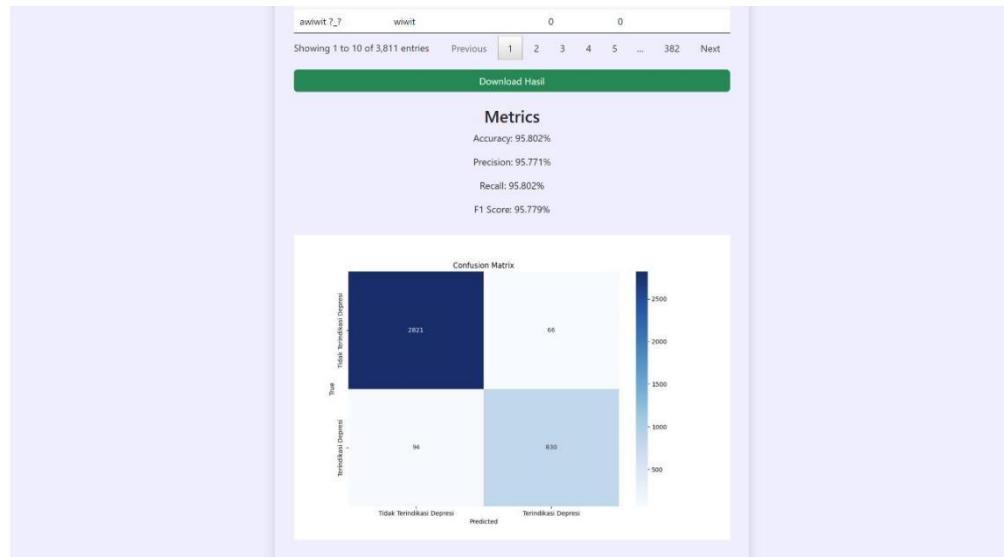
menggunakan *confusion matrix* dibawah tabel. Tampilan halaman testing dipaparkan pada gambar 4.3, 4.4, dan 4.5



Gambar 4.3 Halaman *Testing*

Upload File untuk Testing Model			
Text	Preprocessed Text	Label	Label predict
apa apaan paket gue mau abiez	apa apaan paket gue mau amien	0	0
Feladaputa	feladaputa	0	0
Peraihan terpenting dalam hidup adalah menjadi sesuatu, bukan menemukan sesuatu.	peraihan terpenting hidup menjadi bukan menemukan	0	0
Ndue konco kok. nggarakno emosi ae. Eh emang kita ternan ya?	due konco kok nggarakno emosi ae iki emang teman	0	0
Tanyar..Aku ga terima aku kena skizofrenia, aku gamau minum obat, aku kesiksa sama paranoid, takut tidur, takut kehilangan orang orang tersayang, gabisa sosialisasi, aku berkali2 ganti	anyar aku ga terima aku kena skizofrenia aku gama minum obat aku sama parenti takut tidur takut kehilangan orang orang tersayang gabisa sosialisasi aku berkali ganti obat torus ga mempan malah bikin nambah	1	1

Gambar 4.4 Hasil *Testing*



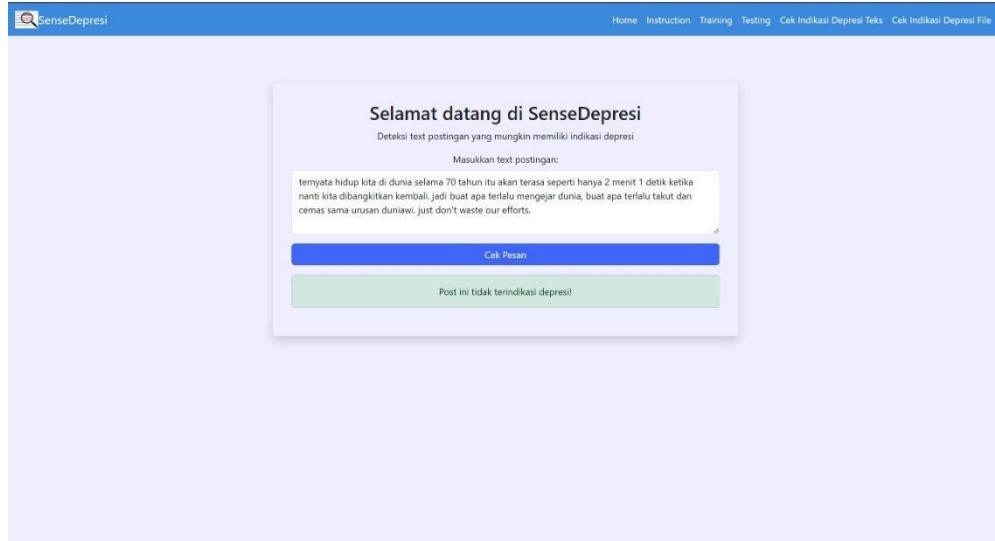
Gambar 4.5 Evaluasi Model menggunakan *Confusion Matrix*

3. Halaman Cek Indikasi Depresi Teks

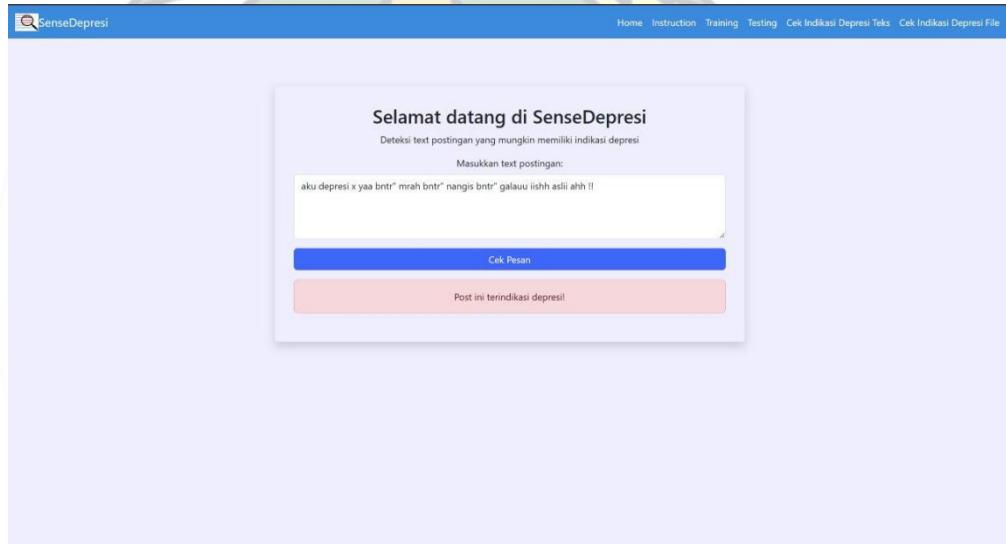
Halaman Cek Indikasi Depresi Teks memungkinkan pengguna untuk melakukan deteksi terhadap postingan teks dengan menginput text tersebut ke dalam *textbox* yang tersedia. Lalu, *input text* akan dideteksi dengan menekan tombol “cek pesan” dan menghasilkan output postingan tidak terindikasi depresi atau postingan terindikasi depresi. Postingan yang tidak terindikasi depresi akan ditampilkan dengan *alert* hijau sedangkan postingan yang terindikasi depresi akan ditampilkan dengan *alert* merah. Pada gambar 4.6, 4.7, dan 4.8 ditampilkan halaman cek depresi teks

The figure shows a screenshot of the SenseDepresi website. At the top, there is a navigation bar with links for 'Home', 'Instruction', 'Training', 'Testing', 'Cek Indikasi Depresi Teks', and 'Cek Indikasi Depresi File'. Below the navigation bar, there is a search bar with placeholder text 'Masukkan text postingan:' and a blue 'Cek Pesan' button. Above the search bar, there is a heading 'Selamat datang di SenseDepresi' and a sub-instruction 'Deteksi text postingan yang mungkin memiliki indikasi depresi'.

Gambar 4.6 Halaman Cek Indikasi Depresi File



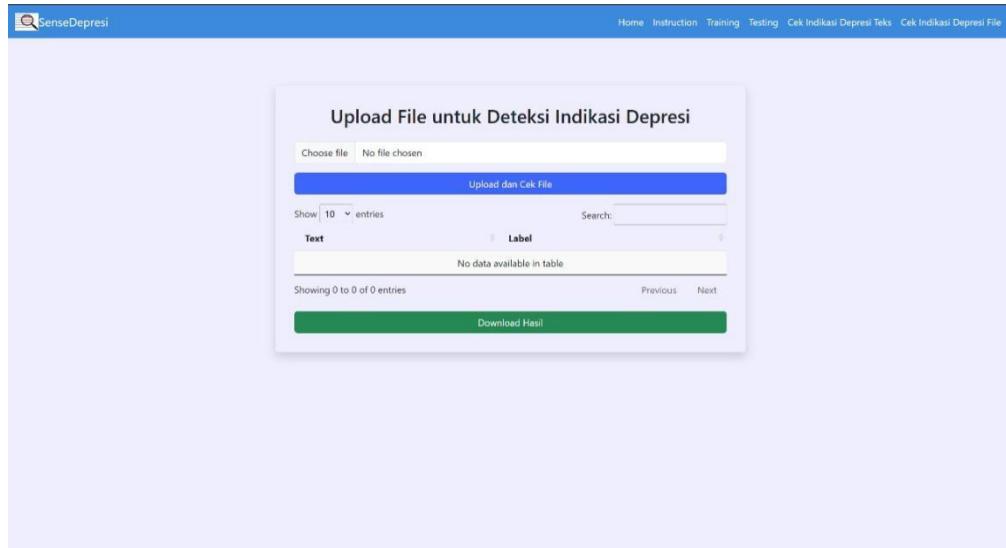
Gambar 4.7 Hasil postingan tidak terindikasi depresi



Gambar 4.8 Hasil postingan terindikasi depresi

4. Halaman Cek Indikasi Depresi File

Halaman Cek Indikasi Depresi File memungkinkan pengguna untuk melakukan deteksi terhadap postingan teks yang berformat file .csv atau .xlsx. File yang diupload akan dilakukan proses prediksi setelah menekan tombol “Cek File”. Hasil dari prediksi akan ditampilkan dalam bentuk tabel yang memuat data teks dan label hasil prediksi dengan nilai 0 atau 1 dimana 0 merepresentasikan postingan tidak terindikasi depresi, sedangkan 1 merepresentasikan postingan terindikasi depresi. Hasil prediksi juga dapat didownload menggunakan tombol “download hasil”. Gambar 4.9 dan 4.10 menunjukkan tampilan halaman cek depresi file.



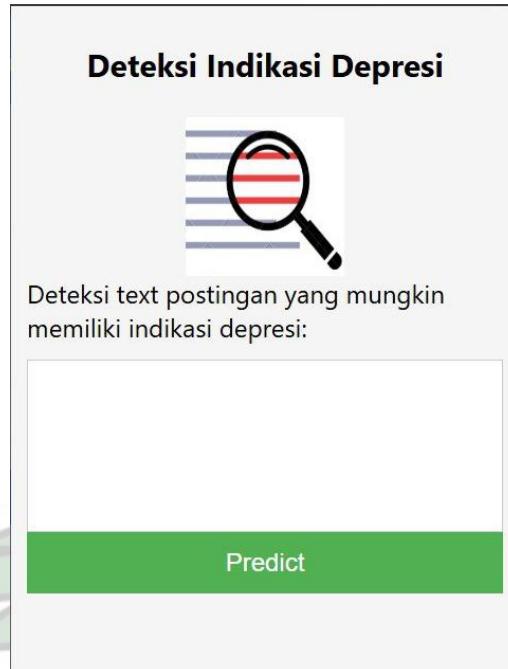
Gambar 4.9 Halaman Cek Indikasi Depresi File

Upload File untuk Deteksi Indikasi Depresi	
Choose file	No file chosen
Upload dan Cek File	
Show 10 entries	Search:
Text	Label
Please jangan maen kompetitif kalau sedang depres	0
Ada yang bisa bantu jawab ini gak? Ada gak sih depres pas hamil? Aku bahagia banget hamil dan sangat bersyukur masya Allah. Tapi beberapa sisi kok aku khawatir takut resah bahkan kadang kalo terbesit di benak. Rasanya gak nyaman panik sesak aku takut bgt	1
pls kawan2 aku depres teruk minggu ni	0
kok gapake profil photo depres ya?	0
Ehh kenapa people people pada depres di bln mei.. persaan w bulan ini chill bgt	0
Hari ini merasa stres dan depres di pekerjaan. Ngak pake cemas sih.	1
bersih2 rumah n kucing semuanya menghilangkan cape n depres terus sleep	0
napa nich depres = debat capres wkwkwk	0
@tanyakan! sebenarnya sama sama pernah	

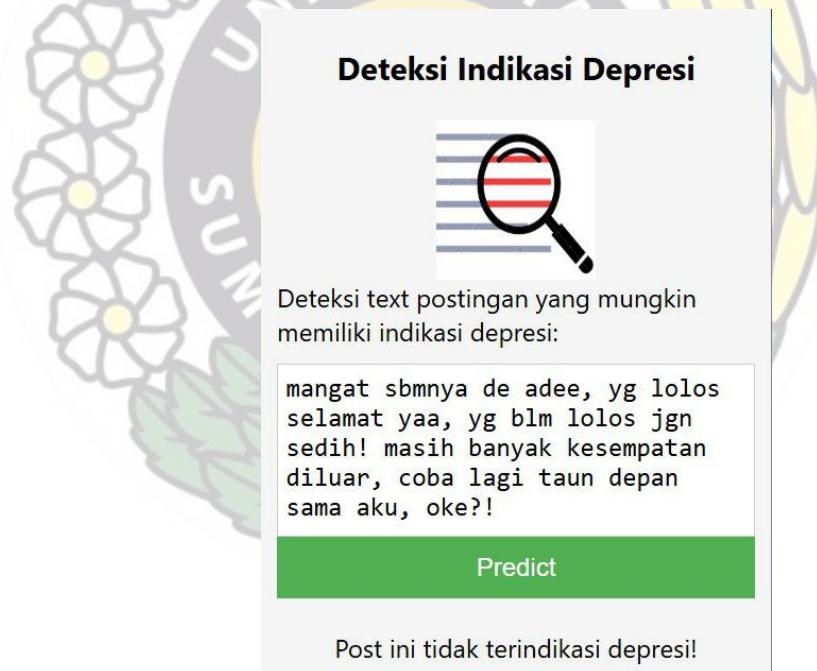
Gambar 4.10 Hasil Cek Indikasi Depresi File

5. Google Chrome Extension

Sistem mendeteksi indikasi depresi juga tersedia dalam bentuk *Google Chrome extension* yang dibuat agar memudahkan pengguna dalam mendeteksi suatu postingan teks secara cepat. Pengguna tidak perlu membuka website dan bisa langsung menggunakan extension yang memiliki cara kerja yang sama dengan halaman cek depresi teks. Tersedia *textbox* untuk menampung input dari pengguna dan tombol “*predict*” untuk memulai proses prediksi. Hasil dari prediksi akan muncul dibawah tombol. Gambar 4.11 dan 4.12 mensajikan tampilan dari *Google Chrome extension*.



Gambar 4.11 *Google Chrome Extension Deteksi Indikasi Depresi*



Gambar 4.12 Hasil Deteksi Indikasi Depresi *Google Chrome Extension*

4.2 Hasil Pelatihan Model

Model yang dikembangkan dalam penelitian ini menggunakan algoritma *Long Short-Term Memory (LSTM)* sebagai base model dengan menggunakan beberapa kombinasi layer agar menghasilkan model yang optimal dalam mendeteksi indikasi depresi. Hasil pengujian kombinasi layers disajikan pada tabel 4.1.

Tabel 4.1 Hasil Pengujian Kombinasi Layers

LSTM	Dropout	LSTM	Dropout	GlobalMaxPooling1D	Dense	Dropout	Dense	Accuracy
64	0.2	64	0.2	-	64	-	1	0.8970
	0.3		0.3	-		-		0.8987
	0.5		0.5	-		-		0.9129
128	0.2	128	0.2	-	64	-	1	0.9038
	0.3		0.3	-		-		0.9038
	0.5		0.5	-		-		0.9046
256	0.2	256	0.2	-	64	-	1	0.9132
	0.3		0.3	-		-		0.9147
	0.5		0.5	-		-		0.9162
64	0.2	-	-	√	64	0.2	1	0.9147
	0.3		-	√		0.3		0.9152
	0.5		-	√		0.5		0.9170
128	0.2	-	-	√	64	0.2	1	0.9195
	0.3		-	√		0.3		0.9167
	0.5		-	√		0.5		0.9173
256	0.2	-	-	√	64	0.2	1	0.9193
	0.3		-	√		0.3		0.9160
	0.5		-	√		0.5		0.9203

Tabel 4.1 Hasil Pengujian Kombinasi Layers (Lanjutan)

LSTM	Dropout	LSTM	Dropout	GlobalMaxPooling1D	Dense	Dropout	Dense	Accuracy
64	0.2	-	-	-	64	-	1	0.9069
	0.3		-	-		-		0.9091
	0.5		-	-		-		0.9099
128	0.2	-	-	-	64	-	1	0.9096
	0.3		-	-		-		0.9048
	0.5		-	-		-		0.9074
256	0.2	-	-	-	64	-	1	0.9033
	0.3		-	-		-		0.9140
	0.5		-	-		-		0.9114
64	0.2	-	-	√	64	-	1	0.9152
	0.3		-	√		-		0.9147
	0.5		-	√		-		0.9132
128	0.2	-	-	√	64	-	1	0.9180
	0.3		-	√		-		0.9173
	0.5		-	√		-		0.9127
256	0.2	-	-	√	64	-	1	0.9271
	0.3		-	√		-		0.9185
	0.5		-	√		-		0.9211

Dari nilai akurasi yang dihasilkan setiap kombinasi, penulis memutuskan untuk menggunakan satu layer LSTM dengan units 256, satu layer *Dropout* dengan nilai 0.2, layer *GlobalMaxPooling1D*, dan dua *dense layer* sebagai arsitektur model LSTM.

4.3 Pengujian Kinerja Model

Model yang telah dilatih menjalani proses pengujian menggunakan 3943 *data testing* yang telah dipisahkan antara *data training* dan *data validation*. *Data testing* akan menjalani tahap *preprocessing* dan *word embedding* seperti yang dilakukan pada tahap *training*. Vektor yang dihasilkan oleh *word embedding* terhadap *data testing* akan digunakan untuk proses deteksi. Hasil pengujian akan ditampilkan pada tabel 4.2 tetapi hanya beberapa data saja yang akan ditampilkan

Tabel 4.2 Hasil Pengujian Kinerja Model

Text	Label Aktual	Label Prediksi	
		0	1
apa apaan paket gue mau abiez	0	0	
Tanyarl..Aku ga terima aku kena skizofrenia, aku gamau minum obat, aku kesiksa sama paranoid, takut tidur, takut kehilangan orang orang tersayang, gabisa sosialisasi, aku berkali2 ganti obat terus ga mempan malah bikin nambah trauma, panik, gelisah Â°Â¥Â~Â°Â¥Â~Â”	1	1	
Jangan salahkan keadaan, dan jangan salahkan takdir. Karna takdir bisa di rubah dan keadaan bisa berputar arah. Jika sedang dalam masalah jgn gelisah percayalah "Allah always with you".	0	0	
Nanti Lebaran dateng ke rumah ya, .Masa iya gak mau salaman .Sama calon pengganti Alberttt:)	0	0	
Nafsu makan turun, Berat badan turun, cemas berlibahan sering nangis apakah ini yang dinamakan depresi	1	1	

Tabel 4.2 Hasil Pengujian Kinerja Model (Lanjutan)

Text	Label Aktual	Label Prediksi
Nggak pernah terbayangkan dalam hidupku, aku ada di situasi yang kayak gini. Bikin aku pingin mati aja.	1	1
Pengen mati tapi takut mati itu gimana? Setiap malam datang aku cemas bgt takut kenapa2, tapi aku juga pengen mati.	1	1
Tak ush ku taut..Tak usah ku bmbang..Tak usah ku ssh..Tak usah ku cema. Ã°Â¥ÂžÂ¶..Yang bisa lanjutin cowo idaman emang, bisa diajak saat teduh bareng Ã°Â¥Â~Â← ÃƒÂ°Ã...Ã¢Â€Â™Ã¢Â„Ã¢ akhir akhir ini	0	0
aku stress bgt, aku depresi dan bnyk masalah, aku ngerasa lelah bgt, semua masalah bikin aku capek, klo aku ngilang sehari gpp kan? aku pgn istirahat seneng seneng take my time alone (cont)	1	1
Aku dulu pernah bilang aku suka bgt sama opening tnmd. Serius. Tidak bohong._x000D_ https://t.co/zMlRFR2yVg	0	0
gue gelisah nungguin ep 20 Ã°Â¥Â¥Â²	0	0
Jangan lupa ucapan terimakasih untuk diri sendiri sebelum tidurÃ¢ÂœÂ„_x000D_ Terima kasih sudah tidak menyerah walau sering merasa kalah wahai diriku sendiri Ã°Â¥Â~Âš	0	0
heal! aku cemas banget sama masa depan aku tiap malem aku overthinking sampai sekarang itu tiap mau tidur aku selalu membayangkan..... membayangkan diri aku lagi berdiri di jembatan dan siap-siap buat loncat :)	1	1

Tabel 4.2 Hasil Pengujian Kinerja Model (Lanjutan)

Text	Label Aktual	Label Prediksi
Bersyukurlah jika ada hati yg gelisah ketika suatu amalan yg sudah biasa dilakukan ditinggalkan. Akan lebih rugi ketika melakukan dosa tapi merasa baik2 saja...	0	0

Hasil pengujian menunjukkan bahwa model bekerja dengan baik dalam mendeteksi postingan teks yang memiliki indikasi depresi dan yang tidak. Model dapat memahami konteks dari postingan teks yang berindikasi depresi dan juga teks yang tidak berindikasi depresi dan mengklasifikasikan postingan tersebut dengan baik. Namun, masih ada beberapa postingan teks yang salah dideteksi oleh model. Tabel 4.3 berisikan data yang memiliki kesalahan label prediksi.

Tabel 4.3 Postingan yang salah dideteksi oleh model dalam testing

Text	Label Aktual	Label Prediksi	Keterangan
gue rasanya mau mati. kalo bunuh diri dibolehin gue mungkij dari dulu udah ngelakuin... tp gue masih mikirin mama ayah yang sayang sama gue... terus juga temen-temen kuliah gue. especially kak aldy & kak nabila dan juga temen-temen virtual gue yg gue sayangin	1	0	Salah
Hallo mamah, beberapa hari ini anak perempuan mu mengalami cemas yang luar biasa, bisakah mama hilangkan rasa ini? Mari jemput saya pulang. _x000D_ Mah, bagaimana kalo aku ga jadi apaÂƒÂ,Â,Â?Â Bagaimana kalo aku memilih untuk mengakhiri kegundahan ini?_x000D_ Maaf mah selalu mematahkan sayapmuÂƒÂ°Â...Â,Â¢Â,,Â¢Â,Â†ÂƒÂ°	1	0	Salah

Tabel 4.3 Postingan yang salah dideteksi oleh model dalam testing (Lanjutan)

Text	Label Aktual	Label Prediksi	Keterangan
Kenapa orang tidak ada yang memahami aku, capek banget sama semua tekanan yang mereka berikan hari ini aku bersyukur banget yaallah, memiliki seseorang yg menjadikan aku like a princess, berbuat hal yg tujuannya hanya ingin buat aku bahagia, dan aku selalu ga pernah nyadarin itu semua, im so sorry but im so thankyou <i>ÃƒÃ¢...Ã“Ã„Ã“</i> film jallikattu (2019) gokil banget dah Ã°Ã¥~Ã„ filmnya seru, ngalir banget, kamera work cakep, openingnya menarik bikin langsung kecantol, cuma hshdhs gatau, w kayak lagi ada di kondisi dimana w gelisah sendiri harus stuck di satu aktivitas dalam waktu lama. im taking my break now.	1 0 0	0 1 1	Salah Salah Salah
Hai cheer me up. Jujur aku udah gatau lagi harus gimana. aku depresi aku ngerasa bukan aku. Cuma bisa ngeliat kosong doang. Tiap malam cuma nangis sampe air mata abis. Dibully, di omongin temen sama saudara sendiri gaenak banget.	1	0	Salah
Aku sedih tp harus bersandar kepada siapa. Harus mengungkapkan kepada siapa dalam wujud manusia. Kalau mengambil saran "bercerita kepada Tuhan", tapi Tuhan tidak bisa mengatakan "tidak apa2" juga tidak bisa memberikan pelukan sebagai tanda bahwa aku tidak sendirian.	1	0	Salah

Tabel 4.3 Postingan yang salah dideteksi oleh model dalam testing (Lanjutan)

Text	Label	Label	Keterangan
	Aktual	Prediksi	
gue depresi, udah gak ada semangat hidup, udh gak tertarik sama apapun, wkt tidur acak2an, dan itu berlangsung lama jg mengganggu aktivitas. is it normal?	1	0	Salah
lompat dari lantai tinggi mungkin pilihan yang bagus	1	0	Salah
Kok ngerasa worry? Kek berasa cemas gt rasa peluru enak sepertinya	0	1	Salah
	1	0	Salah

Kesalahan dalam mendeteksi data tersebut dikarenakan adanya beberapa faktor. Penulis akan melakukan analisis terhadap faktor-faktor yang menyebabkan terjadinya kesalahan dalam proses deteksi. Penjelasan faktor-faktor tersebut dijelaskan sebagai berikut:

1. Teks postingan yang ambigu

Postingan yang memiliki ambiguitas adalah postingan yang memiliki makna yang sulit dipahami tanpa konteks tambahan. Selain itu, postingan yang ambigu ditandai dengan penggunaan kata positif yang berlebihan sehingga membuat model melakukan kesalahan dalam mendeteksi. Contoh kalimat yang ambigu adalah:

- “gue rasanya mau mati. kalo bunuh diri dibolehin gue mungkij dari dulu udah ngelakuin... tp gue masih mikirin mama ayah yang sayang sama gue... terus juga temen-temen kuliah gue. especially kak aldy & kak nabilah dan juga temen-temen virtual gue yg gue sayangin”
- “Kok ngerasa worry? Kek berasa cemas gt ”

Kedua contoh ini merupakan postingan teks yang ambigu dimana contoh pertama menggunakan kata sayang yang memiliki makna positif secara berlebihan, sehingga konteks dari kalimat tersebut lebih condong ke arah positif yang menyebabkan model melakukan kesalahan dalam mendeteksi. Selain itu, contoh kedua adalah kalimat yang ambigu karena kalimat tersebut kurang dalam hal konteks sehingga tidak dapat ditentukan secara jelas apakah postingan tersebut memiliki indikasi depresi atau tidak.

2. Terdapat teks postingan yang memiliki makna tersirat

Selain ambiguitas, terdapat makna tersirat yang tidak ditulis secara eksplisit pada teks postingan yang menyebabkan model kesulitan dalam memahami konteks dari suatu kalimat. Postingan jenis ini ditandai dengan penggunaan kata-kata yang mengindikasikan sesuatu yang dalam dan menggunakan metafora terhadap konteks yang memiliki indikasi depresi. Contoh dari kalimat yang memiliki makna tersirat:

- “rasa peluru enak sepertinya”
- “lompat dari lantai tinggi mungkin pilihan yang bagus”

Kedua contoh tersebut memiliki makna bahwa seseorang ingin melakukan tindakan drastis yaitu bunuh diri yang merupakan tindakan yang dilakukan oleh orang depresi. Tetapi, kalimat tersebut disusun menggunakan kata-kata yang tidak menyebutkan secara eksplisit bahwa orang tersebut ingin melakukan tindakan bunuh diri. Hal ini menyebabkan model melakukan kesalahan dalam proses deteksi dikarenakan kurangnya pembelajaran terhadap postingan teks yang memiliki makna tersirat.

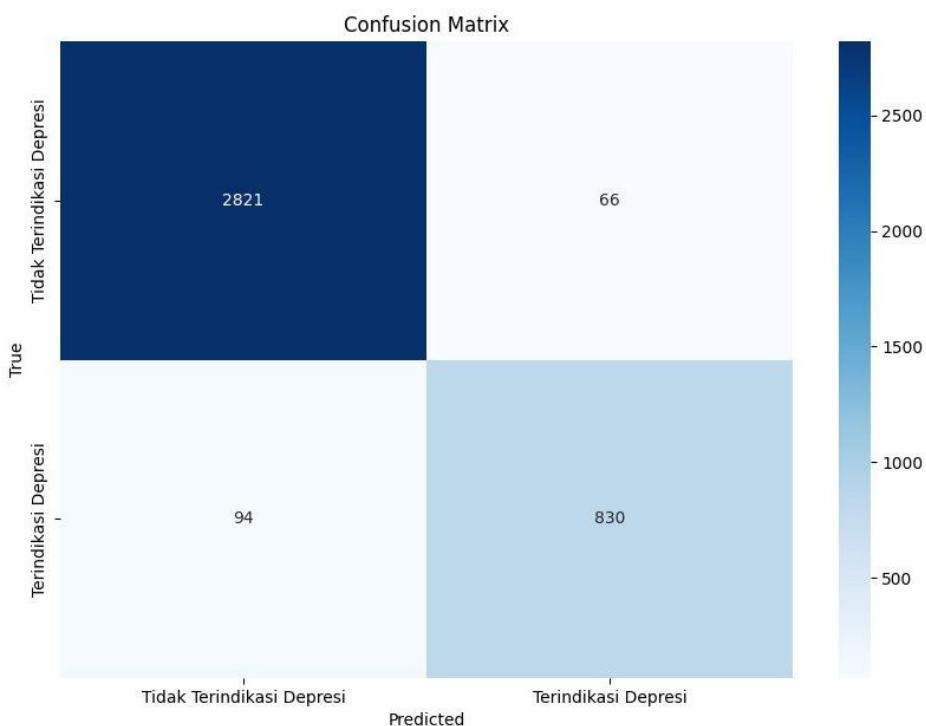
3. Penggunaan kata-kata berbahasa asing

Penggunaan bahasa asing yang dicampur dengan bahasa Indonesia sering kali ditemukan pada teks postingan di platform Twitter (X). Model IndoBERTweet tidak memiliki *vocabulary* untuk kata-kata bahasa asing seperti bahasa Inggris sehingga pada proses *word embedding*, IndoBERTweet gagal dalam memahami konteks kalimat tersebut. Contoh kalimat yang menggunakan bahasa asing:

- “film jallikattu (2019) gokil banget dah Å°Å~Å, filmnya seru, ngalir banget, kamera work cakep, openingnya menarik bikin langsung kecantol, cuma hshdhs gatau, w kayak lagi ada di kondisi dimana w gelisah sendiri harus stuck di satu aktivitas dalam waktu lama. im taking my break now.”
- “hari ini aku bersyukur banget yaallah, memiliki seseorang yg menjadikan aku like a princess, berbuat hal yg tujuannya hanya ingin buat aku bahagia, dan aku selalu ga pernah nyadarin itu semua, im so sorry but im so thankyou ÅfÅ¢...Å“Å,Å”

Dari contoh kalimat di atas, penggunaan kata-kata asing seperti *im taking my break now*, *princess*, dan *im so sorry but im so thankyou* tidak dapat dikenali oleh IndoBERTweet pada proses word embedding sehingga konteks terhadap token-token dari kata-kata tersebut tidak dapat ditetapkan secara tepat. Frasa-frasa tersebut dianggap sebagai frasa yang netral atau negatif sehingga mempengaruhi hasil klasifikasi yang dilakukan oleh model.

Penelitian ini menggunakan *confusion matrix* untuk mengukur kinerja model terhadap performa yang dihasilkan. *Confusion matrix* digunakan agar mendapat visualisasi terhadap data yang dideteksi secara tepat dan tidak tepat. Pada scenario pengujian pertama, tahap preprocessing menghapus emoji secara keseluruhan. Visualisasi *Confusion Matrix* untuk pengujian pertama disajikan pada gambar 4.13.



Gambar 4.13 *Confusion Matrix* Hasil Pengujian Kinerja Model Tanpa Emozi

Dari *confusion matrix* yang telah dipaparkan, sebanyak 2821 data yang tidak terindikasi depresi berhasil dideteksi oleh model dan 66 data yang gagal dideteksi. Selain itu, model juga berhasil mendeteksi 830 data yang terindikasi depresi dan 94 data yang gagal dideteksi. Data nilai dari *Confusion Matrix* pengujian pertama dijabarkan pada tabel 4.4 dan 4.5

Tabel 4.4 Perhitungan data yang tidak terindikasi depresi tanpa emoji

No	Postingan yang tidak terindikasi depresi	Jumlah
1	True Positive	2821
2	True Negative	830
3	False Positive	94
4	False Negative	66

Tabel 4.5 Perhitungan data yang terindikasi depresi tanpa emoji

No	Postingan yang terindikasi depresi	Jumlah
1	True Positive	830
2	True Negative	2821
3	False Positive	66
4	False Negative	94

Data-data pada tabel diatas kemudian digunakan dalam perhitungan nilai *Accuracy*, *Precision*, *Recall*, dan *F-I Score* menggunakan rumus 2.7, 2.8, 2.9, dan 2.10. Nilai-nilai tersebut merupakan nilai metric yang digunakan sebagai penilaian seberapa efektif suatu model.

Perhitungan *Precision*, *Recall*, dan *F-I Score* untuk teks yang tidak terindikasi depresi:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% = \frac{2821}{2821 + 94} \times 100\% = 96.77\%$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% = \frac{2821}{2821 + 66} \times 100\% = 97.71\%$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% = 2 \times \frac{0.9677 \times 0.9771}{0.9677 + 0.9771} \times 100\% = 97.23\%$$

Perhitungan *Precision*, *Recall*, dan *F-I Score* untuk teks yang terindikasi depresi:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% = \frac{830}{830 + 66} \times 100\% = 92.63\%$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% = \frac{830}{830 + 94} \times 100\% = 89.82\%$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% = 2 \times \frac{0.9263 \times 0.8982}{0.9263 + 0.8982} \times 100\% = 91.20\%$$

Perhitungan akurasi untuk keseluruhan data:

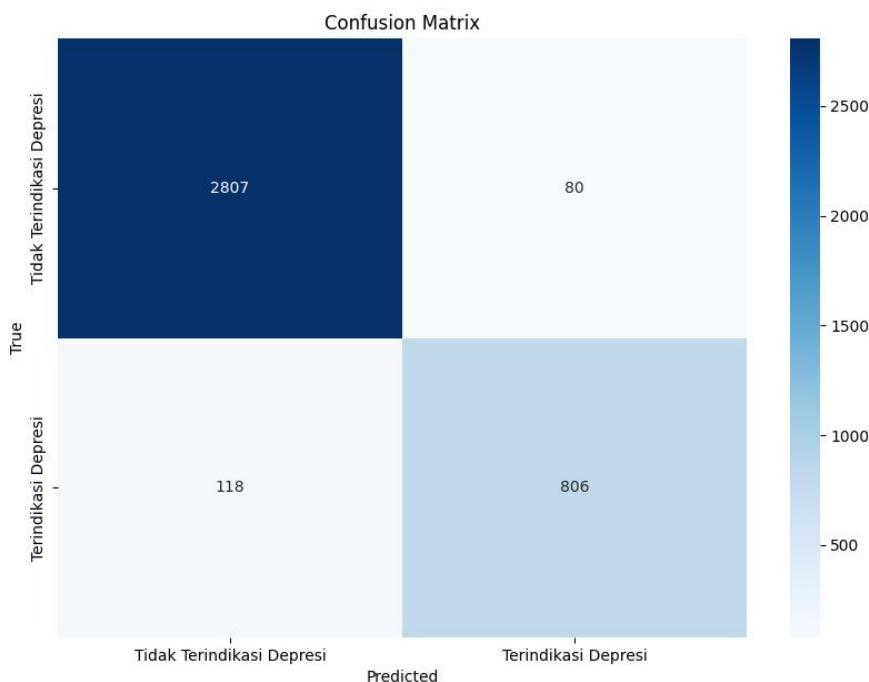
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% = \frac{2821 + 830}{2821 + 830 + 94 + 66} \times 100 = 95.80\%$$

Perhitungan yang telah dilakukan terhadap nilai *metric* teks yang terindikasi depresi dan tidak terindikasi depresi akan dipaparkan pada tabel 4.6

Tabel 4.6 Hasil Komputasi Nilai Evaluasi tanpa emoji

	Precision	Recall	F1-Score
Postingan yang tidak terindikasi depresi	96.77%	97.71%	97.23%
Postingan yang terindikasi depresi	92.63%	89.82%	91.20%

Pada scenario pengujian kedua, tahap preprocessing tidak menghapus emoji, melainkan mengubah emoji menjadi teks yang akan termasuk dalam proses deteksi. Visualisasi *Confusion Matrix* pengujian kedua dipaparkan pada gambar 4.14.



Gambar 4.14 *Confusion Matrix* Hasil Pengujian Kinerja Model Menggunakan Emoji

Dari *confusion matrix* pengujian kedua, sebanyak 2807 data yang tidak terindikasi depresi berhasil dideteksi oleh model dan 80 data yang gagal dideteksi. Selain itu, model juga berhasil mendeteksi 806 data yang terindikasi depresi dan 118

data yang gagal dideteksi. Data nilai dari *Confusion Matrix* pengujian kedua dijabarkan pada tabel 4.7 dan 4.8

Tabel 4.7 Perhitungan data yang tidak terindikasi depresi menggunakan emoji

No	Postingan yang tidak terindikasi depresi	Jumlah
1	True Positive	2807
2	True Negative	806
3	False Positive	118
4	False Negative	80

Tabel 4.8 Perhitungan data yang terindikasi depresi menggunakan emoji

No	Postingan yang terindikasi depresi	Jumlah
1	True Positive	806
2	True Negative	2807
3	False Positive	80
4	False Negative	118

Data-data pada tabel 4.7 dan 4.8 digunakan dalam perhitungan nilai *Accuracy*, *Precision*, *Recall*, dan *F-1 Score* menggunakan rumus 2.7, 2.8, 2.9, dan 2.10. Perhitungan *Precision*, *Recall*, dan *F-1 Score* untuk teks yang tidak terindikasi depresi:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% = \frac{2807}{2807 + 118} \times 100\% = 95.96\%$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% = \frac{2807}{2807 + 80} \times 100\% = 97.22\%$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% = 2 \times \frac{0.9596 \times 0.9722}{0.9596 + 0.9722} \times 100\% = 96.58\%$$

Perhitungan *Precision*, *Recall*, dan *F-1 Score* untuk teks yang terindikasi depresi:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% = \frac{806}{806 + 80} \times 100\% = 90.97\%$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% = \frac{806}{806 + 118} \times 100\% = 87.22\%$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% = 2 \times \frac{0.9097 \times 0.8722}{0.9097 + 0.8722} \times 100\% = 89.05\%$$

Perhitungan akurasi untuk keseluruhan data:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% = \frac{2807 + 806}{2807 + 806 + 118 + 80} \times 100 = 94.80\%$$

Perhitungan yang telah dilakukan terhadap nilai *metric* teks yang terindikasi depresi dan tidak terindikasi depresi akan dipaparkan pada tabel 4.9

Tabel 4.9 Hasil Komputasi Nilai Evaluasi menggunakan emoji

	Precision	Recall	F1-Score
Postingan yang tidak terindikasi depresi	95.96%	97.22%	96.58%
Postingan yang terindikasi depresi	90.97%	87.22%	89.05%

Perbandingan antara evaluasi model tanpa emoji dan menggunakan emoji adalah akurasi yang dihasilkan oleh model tanpa emoji sebesar 95.80% dan akurasi yang dihasilkan oleh model menggunakan emoji sebesar 94.80%.

4.4 Hasil Evaluasi Pakar

Sistem deteksi indikasi depresi yang telah dibangun perlu dievaluasi oleh pakar psikologi untuk memastikan sistem telah bekerja dengan benar. Oleh karena itu, aspek-aspek berikut merupakan penilaian yang digunakan pada evaluasi kepada pakar:

1. Kepuasan (Satisfaction)

Pakar menilai tingkat kepuasan terhadap desain antarmuka yang diimplementasikan pada sistem, kecepatan sistem, dan kemudahan dalam menggunakan website secara keseluruhan

2. Keandalan (Reliability)

Pakar menilai seberapa akurat dan konsisten hasil yang diberikan oleh sistem dalam mendeteksi indikasi depresi.

Pertanyaan yang diberikan menerapkan skala likert sebagai alat ukur penilaian.

Pertanyaan-pertanyaan yang diberikan kepada pakar disajikan pada tabel 4.10

Tabel 4.10 Tabel pertanyaan pada evaluasi pakar

No	Pertanyaan
1	Hasil deteksi yang ditampilkan oleh sistem mudah dipahami.
2	Saya puas dengan kecepatan respon sistem.

Tabel 4.10 Tabel pertanyaan pada evaluasi pakar (Lanjutan)

No	Pertanyaan
3	Sistem ini memberikan hasil yang akurat dalam mendeteksi teks yang terindikasi depresi
4	Sistem ini mampu memahami konteks teks yang terindikasi depresi
5	Sistem ini konsisten dalam memberikan hasil yang benar dalam mendeteksi indikasi depresi dalam suatu teks.
6	Sistem ini mampu digunakan untuk membedakan teks yang berindikasi depresi dan tidak berindikasi depresi berdasarkan beberapa sampel yang digunakan sebagai perwakilan
7	Saya percaya bahwa hasil yang diberikan oleh sistem dapat diandalkan

Evaluasi pakar dilakukan oleh Melva Dumasari Tobing, M.Psi., Psikolog, Linda Gunawan, S.Psi, M.Psi., dan Michael Julian Feno, M.Psi., Psikolog. Hasil dari evaluasi pakar yang telah dilakukan dicantumkan pada tabel 4.11 dengan penilaian Sangat Tidak Setuju, Tidak Setuju, Netral, Setuju, dan Sangat Setuju.

Tabel 4.11 Hasil dari evaluasi pakar

No	Pertanyaan	Melva Dumasari Tobing	Linda Gunawan	Michael Julian Feno
1	Hasil deteksi yang ditampilkan oleh sistem mudah dipahami.	Setuju	Setuju	Sangat Setuju
2	Saya puas dengan kecepatan respon sistem.	Setuju	Setuju	Netral
3	Sistem ini memberikan hasil yang akurat dalam mendeteksi teks yang terindikasi depresi	Setuju	Sangat Setuju	Setuju

Tabel 4.11 Hasil dari evaluasi pakar (Lanjutan)

No	Pertanyaan	Melva Dumasari Tobing	Linda Gunawan	Michael Yulian Feno
4	Sistem ini mampu memahami konteks teks yang terindikasi depresi	Setuju	Setuju	Setuju
5	Sistem ini konsisten dalam memberikan hasil yang benar dalam mendeteksi indikasi depresi dalam suatu teks.	Setuju	Setuju	Setuju
6	Sistem ini mampu digunakan untuk membedakan teks yang berindikasi depresi dan tidak berindikasi depresi berdasarkan beberapa sampel yang digunakan sebagai perwakilan	Setuju	Sangat Setuju	Sangat Setuju
7	Saya percaya bahwa hasil yang diberikan oleh sistem dapat diandalkan	Setuju	Sangat Setuju	Setuju

Berdasarkan hasil evaluasi dari pakar psikologi, sistem yang dikembangkan menghasilkan tingkat kepuasan yang menyatakan sistem sudah mampu melakukan deteksi indikasi depresi dengan baik. Tetapi, dari hasil evaluasi juga mengindikasikan sistem memiliki ruang untuk peningkatan terutama dalam kecepatan respon sistem dan desain antarmuka yang lebih menarik.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian ini, simpulan terhadap implementasi *Long Short-Term Memory (LSTM)* dan IndoBERTweet *Embedding* dalam mendeteksi indikasi depresi dapat dijabarkan sebagai berikut:

1. Algoritma *Long Short-Term Memory (LSTM)* dan IndoBERTweet *Embedding* memberikan tingkat efektivitas yang baik dalam mendeteksi indikasi depresi teks dengan nilai akurasi 95.80% untuk model yang tidak menggunakan emoji.
2. Peforma dari model yang dibentuk menggunakan algoritma *Long Short-Term Memory (LSTM)* bervariasi berdasarkan arsitektur layer yang digunakan. Model yang paling optimal dihasilkan menggunakan arsitektur *layer LSTM* dengan *units* 256, *layer dropout* 0.2, *layer GlobalMaxPooling1D()*, dan 2 *layer dense* dengan masing-masing *units* 64 dan 1.
3. Penggunaan emoji dalam proses mendeteksi indikasi depresi menggunakan algoritma dan arsitektur yang sama menghasilkan akurasi sebesar 94.80%
4. Teks postingan yang masih bersifat ambigu atau memiliki makna tersirat, menggunakan kata-kata bahasa asing, serta keterbatasan data dalam konteks dan variasi kalimat dalam proses training memiliki dampak yang besar dalam kinerja model dalam proses deteksi indikasi depresi

5.2 Saran

Adapun sejumlah saran dari penulis terkait dengan pengembangan kinerja sistem dalam penelitian-penelitian selanjutnya:

1. Penelitian ini menggunakan 12703 data namun masih memiliki kendala dalam mendeteksi beberapa variasi teks yang diakibatkan karena kurangnya varian

data dalam proses training. Karena itu, perlu ditambahkan beberapa variasi dataset untuk meningkatkan kinerja model.

2. Menambahkan jumlah dan varian kata dalam kamus normalisasi serta menambah daftar *stopword* untuk memberikan hasil yang lebih optimal dalam tahap *preprocessing*. Hal ini perlu diperhatikan karena dapat mempengaruhi data teks secara drastis dan meningkatkan pemahaman model dalam proses deteksi.
3. Melakukan perbandingan dan analisa terhadap jumlah tweets yang terindikasi depresi pada tahun-tahun tertentu. Analisa ini dapat membantu mengidentifikasi hal apa saja yang terjadi pada kurun waktu tertentu yang dapat menyebabkan indikasi depresi tinggi.
4. Diharapkan untuk mengimplementasikan kombinasi metode algoritma yang berbeda supaya dibandingkan dengan hasil dari kombinasi metode algoritma *Long Short-Term Memory (LSTM)* dan IndoBERTweet *Embedding* untuk penelitian selanjutnya.

DAFTAR PUSTAKA

- Amanat, A., Rizwan, M., Javed, A. R., Abdelhaq, M., Alsaqour, R., Pandya, S., & Uddin, M. (2022). *Deep Learning for Depression Detection from Textual Data*. 1–13.
- Azahra, N. M., & Setiawan, E. B. (2023). Sentence-Level Granularity Oriented Sentiment Analysis of Social Media Using Long Short-Term Memory (LSTM) and IndoBERTweet Method. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI)*, 9(1), 85–95. <https://doi.org/10.26555/jiteki.v9i1.25765>
- Bello, A., Ng, S. C., & Leung, M. F. (2023). A BERT Framework to Sentiment Analysis of Tweets. *Sensors*, 23(1). <https://doi.org/10.3390/s23010506>
- Birunda, S. S., & Devi, R. K. (2020). A Review on Word Embedding Techniques for Text Classification. In J. S. Raj, A. M. Liliyasu, R. Bestak, & Z. A. Baig (Eds.), *International Big Data and IoT Computer Networks, Conference on (ICCBI - 2019)* (Vol. 10, pp. 162–168).
- Caelen, O. (2017). *A Bayesian interpretation of the confusion matrix*. September, 429–450. <https://doi.org/10.1007/s10472-017-9564-8>
- Devika, S. P., Pooja, M. R., Arpittha, M. S., & Vinayakumar, R. (2023). BERT-Based Approach for Suicide and Depression Identification. *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022*, 612, 435–444. https://doi.org/10.1007/978-981-19-9228-5_47
- El-Ramly, M., Alshaer, G., Abu-Elyazid, H., Adib, N., El-Shazly, M., Mo'men, Y., & Eldeen, K. A. (2021). CairoDep : Detecting Depression in Arabic Posts Using BERT Transformers. *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 207–212. <https://doi.org/10.1109/ICICIS52592.2021.9694178>
- Elnagar, A., Yagi, S., Mansour, Y., Lulu, L., & Fareh, S. (2023). A benchmark for evaluating Arabic contextualized word embedding models. *Information Processing and Management*, 60(5), 103452.
- Fadhel, M., & Maharani, W. (2024). Depression Detection of Users in Social Media X using IndoBERTweet. *Sinkron*, 8(2), 885–891. <https://doi.org/10.33395/sinkron.v9i2.13354>

- Fine, A. B., Crutchley, P., Blase, J., Carroll, J., & Coppersmith, G. (2020). *Assessing population-level symptoms of anxiety , depression , and suicide risk in real time using NLP applied to social media data.* 50–54.
- Hidayat, I. R., & Maharani, W. (2022). *General Depression Detection Analysis Using IndoBERT Method.* 8(1), 41–51. <https://doi.org/10.21108/ijoict.v8i1.634>
- Jain, P., Srinivas, K. R., & Vichare, A. (2022). *Depression and Suicide Analysis Using Machine Learning and NLP.* <https://doi.org/10.1088/1742-6596/2161/1/012034>
- Kadhim, A. I. (2018). An Evaluation of Preprocessing Techniques for Text Classification. *International Journal of Computer Science and Information Security*, 16(6), 22–32. <https://sites.google.com/site/ijcsis/>
- Koto, F., Lau, J. H., & Baldwin, T. (2021). INDOBERTWEET: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 10660–10668. <https://doi.org/10.18653/v1/2021.emnlp-main.833>
- Leis, A., Mayer, M., Ronzano, F., & Torrens, M. (2020). *Clinical-Based and Expert Selection of Terms Related to Depression for Twitter Streaming and Language Analysis.* 0. <https://doi.org/10.3233/SHTI200296>
- Lorenzoni, G., Tavares, C., Nascimento, N., Alencar, P., & Cowan, D. (2024). *Assessing ML Classification Algorithms and NLP Techniques for Depression Detection : An Experimental Case Study.*
- Manaswi, N. K. (2018). Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras. In *Apress*. <https://www.hlevkin.com/hlevkin/45MachineDeepLearning/DL/Deep Learning with Applications Using Python.pdf>
- Nordgaard, J., Sass, L. A., & Parnas, J. (2013). The psychiatric interview: Validity, structure, and subjectivity. *European Archives of Psychiatry and Clinical Neuroscience*, 263(4), 353–364. <https://doi.org/10.1007/s00406-012-0366-z>
- Riziana, K. F., & Darmawan, A. (2023). *Hubungan tingkat gejala depresi dengan ide bunuh diri pada remaja sekolah menengah atas.* 39–47.
- Salman, U. T., Rehman, S., Alawode, B., & Alhems, L. M. (2021). Short Term Prediction of Wind Speed Based on Long-Short Term Memory Networks. *FME Transactions*, 49(3), 643–652. <https://doi.org/10.5937/fme2103643S>

- Santra, A. K., & Christy, C. J. (2012). *Genetic Algorithm and Confusion Matrix for Document Clustering 1*. 9(1), 322–328.
- Sikström, S., Kelmendi, B., & Persson, N. (2023). *Assessment of depression and anxiety in young and old with a question-based computational language approach*. 1–10. <https://doi.org/10.1038/s44184-023-00032-z>
- Singh, J., Wazid, M., Singh, D. P., & Pundir, S. (2022). An embedded LSTM based scheme for depression detection and analysis. *Procedia Computer Science*, 215, 166–175. <https://doi.org/10.1016/j.procs.2022.12.019>
- Singh, P., Singh, N., Singh, K. K., & Singh, A. (2021). Chapter 5. Diagnosing of disease using machine learning. In *Machine Learning and the Internet of Medical Things in Healthcare*. INC. <https://doi.org/10.1016/B978-0-12-821229-5.00003-3>
- Smith, K. M., Renshaw, P. F., & Bilello, J. (2013). The diagnosis of depression : current and emerging methods. *Comprehensive Psychiatry*, 54(1), 1–6. <https://doi.org/10.1016/j.comppsych.2012.06.006>
- Yang, C. chen, Holden, S. M., & Ariati, J. (2021). Social Media and Psychological Well-Being Among Youth: The Multidimensional Model of Social Media Use. *Clinical Child and Family Psychology Review*, 24(3), 631–650. <https://doi.org/10.1007/s10567-021-00359-z>